Mimicry of a conceptual hydrological model (HBV): what's in a name?

Jansen, Koen F ; Teuling, Adriaan J ; Craig, James R ; Dal Molin, Marco ; Knoben, Wouter J M ; Parajka, Juraj ; Vis, Marc ; Melsen, Lieke A

Abstract: Models that mimic an original model might have a different model structure than the original model, that affects model output. This study assesses model structure differences and their impact on output by comparing 7 model implementations that carry the name HBV. We explain and quantify output differences with individual model structure components at both the numerical (e.g., explicit/implicit scheme) and mathematical level (e.g., lineair/power outflow). It was found that none of the numerical and mathematical formulations of the mimicking models were (originally) the same as the benchmark, HBV-light. This led to small but distinct output differences in simulated streamflow for different numerical implementations (KGE difference up to 0.15), and major output differences due to mathematical differences (KGE median loss of 0.27). These differences decreased after calibrating the individual models to the simulated streamflow of the benchmark model. We argue that the lack of systematic model naming has led to a diverging concept of the HBV-model, diminishing the concept of model mimicry. Development of a systematic model naming framework, open accessible model code and more elaborate model descriptions are suggested to enhance model mimicry and model development.

DOI: https://doi.org/10.1029/2020wr029143

Originally published at:
Jansen, Koen F; Teuling, Adriaan J; Craig, James R; Dal Molin, Marco; Knoben, Wouter J M; Parajka, Juraj; Vis, Marc; Melsen, Lieke A (2021). Mimicry of a conceptual hydrological model (HBV): what's in a name? Water Resources Research, 57(5):e2020WR029143.
DOI: https://doi.org/10.1029/2020wr029143

AGU ADVANCING EARTH AND SPACE SCIENCE

Water Resources Research

RESEARCH ARTICLE
10.1029/2020WR029143

Mimicry of a Conceptual Hydrological Model (HBV): What's in a Name?

Koen F. Jansen[1], Adriaan J. Teuling[1], James R. Craig[2], Marco Dal Molin[3,4,5], Wouter J. M. Knoben[6], Juraj Parajka[7], Marc Vis[8], and Lieke A. Melsen[1]

[1]Hydrology and Quantitative Water Management, Wageningen University, Wageningen, The Netherlands, [2]Civil and Environmental Engineering Department, University of Waterloo, Waterloo, Ontario, Canada, [3]Department Systems Analysis, Integrated Assessment and Modelling, Eawag, Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland, [4]The Centre of Hydrogeology and Geothermics, University of Neuchâtel, Neuchâtel, Switzerland, [5]Department of Water Resources and Drinking Water, Eawag, Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland, [6]Centre for Hydrology, University of Saskatchewan, Canmore, Alberta, Canada, [7]Institute of Hydraulic Engineering and Water Resources Management, Technische Universität Wien, Vienna, Austria, [8]Department of Geography, University of Zurich, Zürich, Switzerland

Key Points:

- The concept of model mimicry is evaluated by comparing models that bear the same name
- A step-wise comparison of model components against the benchmark model revealed a simulated outflow difference of up to 0.15 point in the Kling-Gupta efficiency metric for numerical differences
- Model structures are often difficult to access and need a systematic model naming framework to advance model development

Correspondence to:

L. A. Melsen and K. F. Jansen,
lieke.melsen@wur.nl
koenatleet@hotmail.com

Citation:

Jansen, K. F., Teuling, A. J., Craig, J. R., Dal Molin, M., Knoben, W. J. M., Parajka, J., et al. (2021). Mimicry of a conceptual hydrological model (HBV): What's in a name? *Water Resources Research*, *57*, e2020WR029143. https://doi.org/10.1029/2020WR029143

Received 5 NOV 2020
Accepted 19 APR 2021

Abstract Models that mimic an original model might have a different model structure than the original model, that affects model output. This study assesses model structure differences and their impact on output by comparing 7 model implementations that carry the name HBV. We explain and quantify output differences with individual model structure components at both the numerical (e.g., explicit/implicit scheme) and mathematical level (e.g., lineair/power outflow). It was found that none of the numerical and mathematical formulations of the mimicking models were (originally) the same as the benchmark, HBV-light. This led to small but distinct output differences in simulated streamflow for different numerical implementations (KGE difference up to 0.15), and major output differences due to mathematical differences (KGE median loss of 0.27). These differences decreased after calibrating the individual models to the simulated streamflow of the benchmark model. We argue that the lack of systematic model naming has led to a diverging concept of the HBV-model, diminishing the concept of model mimicry. Development of a systematic model naming framework, open accessible model code and more elaborate model descriptions are suggested to enhance model mimicry and model development.

1. Introduction

A growing global population and climate change pose many threats to natural resources, including global freshwater resources (Wagener et al., 2010). In order to better understand and quantify the impacts of current and projected global developments, models are necessary. These models can predict and answer "what if" questions by running multiple scenarios, thereby providing insight into past changes and allowing preparation for changes to come (Beven, 2011; Scibek & Allen, 2006; Teuling et al., 2019; Zektser & Loaiciga, 1993). The quality of the model is key to reliable simulations, and is determined by our ability to understand the system and represent the dominant processes of the system appropriately (Clark, Kavetski, & Fenicia, 2011; Kirchner, 2006). In hydrology, these processes vary between catchments and climate conditions. Not surprisingly, it has been found that model choice can affect not only the magnitude but even the direction of trends in streamflow as a result of projected climate change (Melsen, Addor, et al., 2018). Therefore, each study requires careful model selection based on catchment understanding and the goal of the study (Addor & Melsen, 2019).

In theory, different models provide different results for the same input (for a sufficiently varied range of inputs activating the differing model components). Following this rationale, it would be expected that similar models give similar results. It might be easier to study if and why similar models behave differently, than why different models behave similarly. To define similarity, we make use of the concept "model mimicry" (Clark, Nijssen, et al., 2015a). The term mimicry stems from biology, where animals mimic one another—for instance to increase their chance of survival. In order to define mimicry in a hydrological model context, it is useful to consider the different stages in model development (Figure 1).

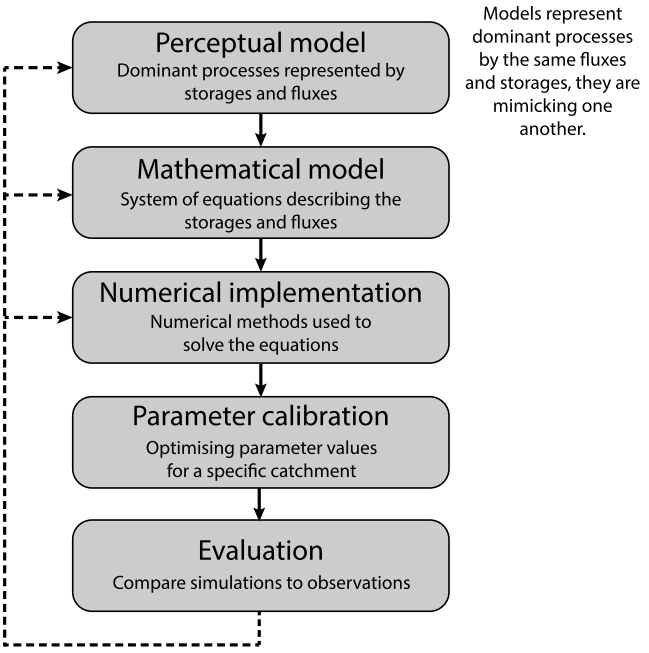


Figure 1. Stages of conceptual model development. Here, model mimicry starts when the perceptual model is the same. The stages are based on Beven (2011) and Clark & Kavetski, 2010; Gupta, Clark, et al., 2012, (note that their definitions are slightly different). Each box indicates a stage.

Model development is typically considered to consist of five stages, starting with a perceptual model and ending with model evaluation. The perceptual model is a broad description of how the system works and what the important process are, for example, there are two main aquifers. This perception is elaborated in a mathematical model, describing the storages and fluxes between those storages in mathematical formulas, e.g., the change of the upper storage is described by $\frac{dS_4}{dt} = infiltration - k_1 * S_4 - percolation$ for the HBV-6-model with k_1 being the outflow coefficient and S_4 the stage in the upper soil zone.

The next stage is the numerical implementation which applies numerical methods to solve the equations, for example, using an implicit Euler scheme. Those three stages form the model structure. After that, the model is often calibrated, to determine and optimize parameter values for a specific catchment, and evaluated, to compare model results with observations. This comparison can be satisfactory (similar enough to observed data), but generally results in new insights that need model adjustments and/or hypothesis testing, thus restarting the development cycle at one of the previous stages depending on the adjustment.

Based on these five stages, we define model mimicry: *model mimicry is the concept of reproducing a model such that it has the same underlying perceptual model, but a possibly different mathematical or numerical implementation as the original model*. If modelers have the same system understanding and agree over dominant processes, their perception of the system is the same. The same processes are included in the model; the output is expected to be similar and the models are mimicking one another. The actual implementation of the perceptual model, however, can differ. When the mathematical model is also the same, the output is expected to be nearly identical. However, Clark and Kavetski (2010) already showed that this is not always the case. They applied eight different numerical time stepping algorithms to six different model structures within FUSE (Clark, Slater, et al., 2008, Framework for Understanding Structural Errors) and found that the numerical implementation could have major impact on model results and could mask model structure errors. Mimicking models might thus differ in their mathematical and numerical implementation, and this might lead to differences in model output.

This study is restricted to model mimicry, which is distinctively different from model emulation which we define as *the practice of simplifying an existing complex model such that it produces similar results while using less computation power*. For model emulation, the model structure could be completely changed, as long as the output is similar. In this way, the focus is on increasing computation efficiency in order to make a wider range of applications feasible, like larger scales, sensitivity analysis, and uncertainty analysis (Gladish et al., 2018). Model mimicry is more fundamental and focuses on understanding the relation between model structure and model output.

Models are often mimicked when a mathematical model is presented but the code is not available or not open source (Weiler & Beven, 2015). In particular, commonly used conceptually simple models such as TOPMODEL and HBV, have regularly been reproduced (Bergström, 2006; Peters et al., 2003). Though purely mimicking variants are often found, some have added an extra model component, thereby changing the perceptual model. This poses the question whether this should still be considered as model mimicry. This is for instance the case for the study of Uhlenbrook et al. (1999), who compared three existing variants of HBV with a slightly different perceptual model and added four other model structures to find the best performing version. The variants vary in number of land classes, distributed and lumped parameters, number of reservoirs, and type of lag functions. Similarly, Girons Lopez et al. (2020) compared a wide variety of snow representations for the HBV-model in a mountainous area in Central Europe. Even though the perceptual models differ in both studies, they all carry the name "HBV" implying that those versions are mimicking

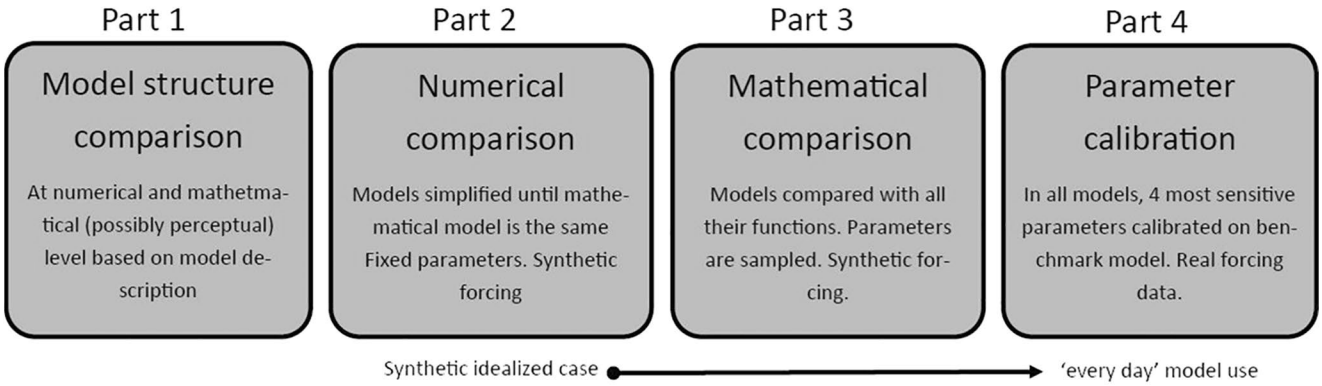
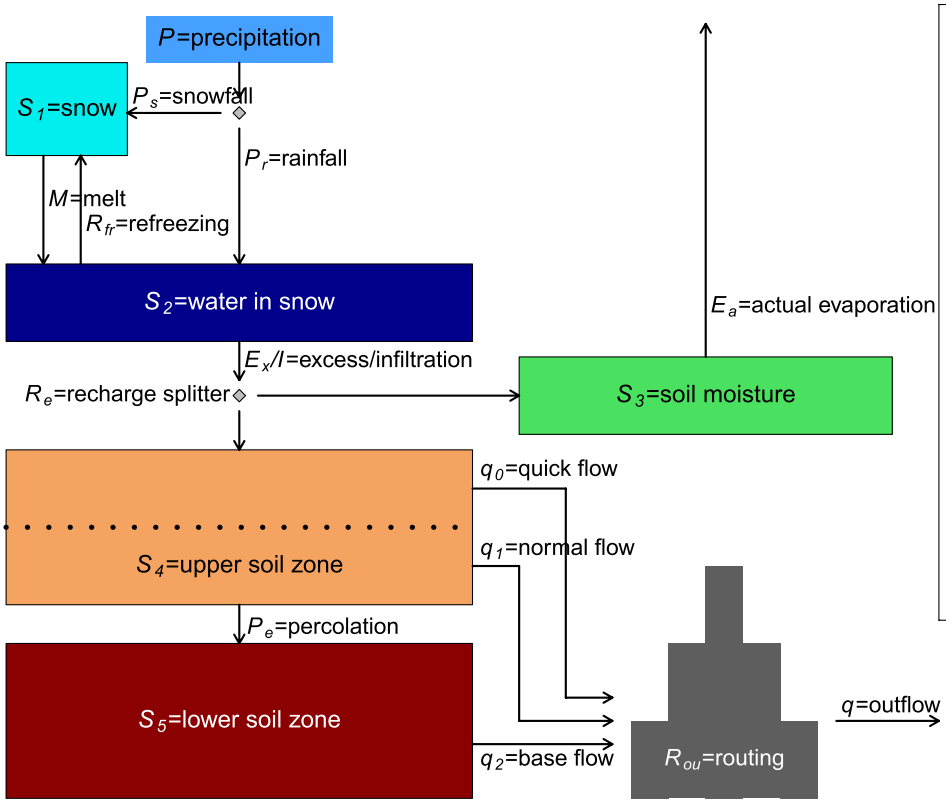


Figure 2. Structure of the current study. Model mimicry is evaluated in four steps moving from simplified to common model use while explaining model output differences. The first part evaluates model structure differences and those differences are used in the other three parts to explain output differences. In Part 2 till 4, model output comparison moves from simplified idealized modeling to more "every day" model use. In Part 2, model complexity is step-by-step increased to isolate the effect of the numerical implementations on model output. In Part 3, mathematical model differences are evaluated and parameters are sampled. In Part 4, the models are calibrated, thereby reflecting more "every day" model use.

the original version and the output is, to some extent, similar. However, not all components of the original version are mimicked and the variations on the original version may lead to quite different outputs.

Modular modeling frameworks (MMFs) are promising, relatively new tools in hydrology that aim at mimicking and incorporating several existing models and exchange modular parts, thus allowing hypothesis testing of single model components (Clark, Nijssen, et al., 2015b; Fenicia et al., 2011). Several MMFs have been developed over the last few years for this purpose (Clark, Nijssen, et al., 2015a; Clark, Slater, et al, 2008; Coxon et al., 2019; Craig et al., 2020; Dal Molin et al., 2020; Knoben, Freer, Fowler, et al., 2019; Leavesley et al., 2002). Although mimicry is fundamental for MMFs, so far only the studies of Craig et al. (2020) and Knoben, Freer, Fowler, et al. (2019) documented the capability of their MMFs to mimic existing models. It should be noted that Nijssen et al. (2018) tested model mimicry within SUMMA but these results have not been formally published. Knoben, Freer, Fowler, et al. (2019) based their mimicked model on journal paper documentation and found considerable output differences which they discuss in detail, whereas Craig et al. (2020) was able to reach nearly identical results for six original models for which the code was available. Several MMFs contain mimicked versions of the same model making them suitable tools in the study of model mimicry. The aim of this study is to assess how model mimicry is reflected in model structure similarity and affects simulated outflow differences by models that can reasonably be expected to have similar internal structures.

The aim of this study is to quantify the effectiveness of model mimicry, that is, to investigate the similarities and differences in model output produced by models that can reasonably be expected to have similar internal structures. It intends to contribute to model structure understanding and MMF development by comparing (the performance of) similar model variants, both within and outside MMFs. We will address the following research question: to what extent do models that bear the same name, mimic model structure and simulate the same output? To this end we will evaluate seven models of the HBV-family (models that are based on a previous HBV-version), in four steps (Figure 2). First, we will compare the model structure at the numerical and mathematical level to the benchmark (HBV-light, discussed in the next section). Those model structure differences are used to explain and understand model output differences. Second, we will explore to what extent the numerical implementation affects output differences. To this end, mathematical model differences are resolved by starting from a simple model structure, switching off mathematical differences, and increasing model complexity in a step-wise fashion. The third part of this study looks into the impact of mathematical model differences on model output similarity. Parameters are sampled to allow mathematical model differences. In the fourth and final part, parameters are calibrated to minimize model output differences. This way, the study moves from purely theoretical model settings to "every day" model practice. This approach allows for the separate investigation of numerical implementation and mathematical model differences and their impact on output differences, but also helps to explain and understand those



Condition	Formula
$T \le t_t$	$P_s = P s_{cf}$
$T \le t_t$	$R_{fr} = (t_t - T) c_{fmax} c_{fr}$
$T > t_t$	$P_r = P$
$T > t_t$	$M = (T - t_t) c_{fmax}$
$T > t_t$	$E_x = S_{2old} - (S_1 c_{wh})$
$S_{2old} < S_1 c_{wh}$	$I = P_r + M - \delta S_2$
$S_{2old} \ge S_1 c_{wh}$	$I = P_r + M$
	$R_e = (I + E_x)(\frac{S_3}{F_C})^{beta}$
$S_3 < F_c L_P$	$E_a = E_p \frac{S_3}{F_c L_P}$
$S_3 \ge F_c L_P$	$E_a = E_p$
	$P_e = per$
$S_4 > u_{zl}$	$q_0 = (S_4 - u_{zl}) k_0$
	$q_1 = S_4 k_1$
	$q_2 = S_5 k_2$
	$R_{ou} = \mathrm{f}(maxbas, q)$
	$q = q_1 + q_2 + q_0$

Figure 3. Mathematical model of HBV-6-model. The boxes indicate the storages and arrows indicate the fluxes. Model states are shown in different colors; this color scheme is used in later figures to quickly refer to specific (parts of) model structures. E_p = potential evaporation. The mathematical equations describe the fluxes (and thus water balance for the storages). Parameter description is shown in Appendix A and a full model description can be found in Seibert (2005) and Bergström (1992).

output differences. The resulting insights in differences in model output of models that bear the same name will answer the central question of "what's in a name?"

2. HBV Model

A good example of an often mimicked model is the HBV model. This conceptual hydrological model was first developed in 1973, and later revised to HBV-6 and HBV-96 in 1992 and 1997, respectively (Bergström, 1992; Bergström & Forsman, 1973; Lindström et al., 1997). Many variants of the model have been published since and even more variants can be found at different institutes (Bergström, 2006). Given its widespread use and the many different implementations available, HBV provides an excellent case study for our study.

2.1. Model Description

The HBV model can be split into a snow, soil moisture, response, and routing routine while some variants include lake and glacier parameterizations as well (Seibert, 2005). Most versions have four or five storages and around 14 parameters. The model can be deployed in a (semi-)distributed fashion accounting for elevation and/or vegetation differences. Figure 3 gives an overview of the HBV-6 variant, its parameters and flux formulations.

Precipitation is either diverted to water in snow storage (S_2) as rainfall or to snow storage (S_1) as snowfall based on the temperature being above or below the temperature threshold (t_t). The same threshold governs melt and refreezing, the fluxes between S_1 and S_2. Infiltration takes place when the storage capacity of S_2 is exceeded. Infiltration is split over soil moisture storage (S_3) and the upper soil zone (S_4) based on S_4 and F_C. S_3 empties by evaporation (E_a). S_4 has a constant percolation to the lower soil zone (S_5) and both S_4 and

Table 1
HBV Model-Variants Employed in This Study

Name	Developer	Language	Citations[a]	Type	Lump/Dis	E/I	Seq/Sim
HBV-light	Seibert and Vis (2012)	VB.NET (.exe)	276	Singular	Dis	E/Adaptive[b]	Seq
TUW	Parajka et al. (2007)	R[c]	131	Singular	Dis	E/A[d]	Seq
MAC	Samuel et al. (2011)	Matlab (.exe)	139	Singular	Lump	E?[e]	Seq
EDU1	AghaKouchak and Habib (2010)	Matlab	86	Singular	Lump	E	Seq
EDU2	~	~	~	~	~	~ ~	~
SuperflexPy	Dal Molin et al. (2020)	Python	0[f]	MMF	Dis	I	Seq/Sim
Raven1	Craig et al. (2020)	C++ (.exe)	1	MMF	Dis	E[g]	Seq
Raven2	~	~	~	~	~	~	~
MARRMoT1	Knoben, Freer, Fowler, et al. (2019)	Matlab	7	MMF	Lump	I/E[h]	Sim
MARRMoT2	~	~	~	~	~	~	~

Note. The adapted variants (see Section 3) are shown in gray for the overview, but have no major numerical differences (indicated with a ~). Numerical aspect is shown in the last two columns. Lump/Dis stand for spatially lumped/semi-distributed, E/I for explicit/implicit, and Seq/Sim for sequential/simultaneous. Most numerical information was gathered by looking at the code since the paper description was not always complete. Some codes were not accessible because of an executable wrapped around the code. Therefore, numerical descriptions are also based on model result comparison and are thus discussed in the results. [a]As of June 2020, google scholar. [b]Only 1 mm at a time is handled for the infiltration splitter. If there is 2 mm of precipitation. First, the first 1 mm is handled by the infiltration splitter (splitting infiltration over S_3 and S_4), thereafter the updated S_3 is used to handle the 2nd mm and divide this over S_3 and S_4. Potential evaporation is averaged between beginning and end of the time step. [c]Originally written in FORTRAN, but translated and made available in R as well. The R implementation is used here. [d]Mostly explicit but outflow is calculated with an analytical solution. [e]A complete numerical scheme could not be provided as it misses a numerical description, the code is not accessible, and the only available output is simulated streamflow. Due to its similarity to the other models, it is likely that it has an explicit first order scheme but this cannot be confirmed with certainty. [f]SuperflexPy is a new flexible modeling framework written in Python that embrace the same modeling principles of SUPERFLEX (Fenicia et al., 2011). A pre-release version was used. [g]Craig et al. (2020) mentions the possibility for other schemes (implicit iterative Heun for specific processes) but only the implementation of ordered series and explicit Euler algorithms are currently described in the user's manual. [h]Option for both implicit and explicit scheme. The implicit scheme has been used except for the excess flow. Besides, the melt flow has been changed in MARRMoT2 to make it behave explicitly.

S_5 have a linear outflow (q_1 = normal flow and q_2 = base blow). Besides, a quick flow (q_2) is activated if S_4 exceeds a threshold (u_{zl}). The three flows are added and routed resulting in the outflow (q). An elaborate model description can be found in Bergström (1992) and Seibert (2005) and a parameter description is given in Appendix A.

2.2. Model Variants

A first search of models bearing the name "HBV" or referring to the name "HBV" resulted in six recently developed MMFs that provide building blocks for HBV variants, and nine HBV-based singular models (models with a mostly fixed model structure), both termed as variants throughout this paper. Four singular variants (including the benchmark model) and three MMFs were selected based on availability (open source), number of citations, and programming language (R, Python, MATLAB to easily access and if necessary modify the code). An overview of the selected models is provided in Table 1. The numerical characteristics of these HBV variants are discussed in the results.

HBV-light (Seibert, 2005) is chosen as benchmark model to compare model structure and output. This model variant is suited as benchmark because it very closely resembles the HBV-6 version on both mathematical as well as numerical level, though detailed comparison among these two are not available (Seibert, 2005). All model variants explored in this study can be traced back to either the HBV-6 or the HBV-96 version as the base model, while all models were published more recently. The HBV-96 version deviates from the HBV-6 version in: (1) the outflow of the upper storage—the quick flow and linear flow are replaced by a non-linear flow, (2) addition of capillary rise, (3) snow/rain threshold where the discrete threshold is replaced by a linear interval, and (4) the monthly evaporation. Besides the availability of the model, the HBV-light, and thus HBV-6 version, is favored above HBV-96 because of the less complex process formulation. More complex formulations can often be switched off by choosing the right parameter whereas adding complexity

required adding code lines to the model code, which is not always possible. Thus, the HBV-6 version allows for analysis of more similar model structures.

2.3. Implemented Changes to Original Model Variants

Mathematical modifications have been made to three of the original model variants to make the mathematical model more similar to the benchmark model and aid numerical implementation comparison. The adapted models are treated as different model variants. The original and adapted model are indicated with a 1 (original) or 2 (adapted), respectively, after the model name. The adapted models are MARRMoT2, Raven2, and EDU2. An example is that a quick outflow flux is added to MARRMoT2 and the non-linear *alpha* component is removed (changing the model component from HBV-96 to HBV-6). Thus, MARRMoT2 has an identical mathematical model for simulation of outflow and output differences can be attributed solely to numerical implementation differences. Apart from these three variants, a small change was made to TUW to avoid NA data (discussed in results). Besides mathematical changes, numerical changes were made to avoid model crashes or reduce unrealistic behavior. An example is the infiltration flux calculation in EDU2 which originally led to complex numbers in some cases. A complete overview of the differences between the original and adapted models can be found in Appendix B1.

The singular models only contained one model structure (that was inspired by HBV). For the MMFs, several extra steps had to be taken in order to construct a mimicked HBV. The three MMFs differed considerably in regards to the extent that they were ready-to-use. MARRMoT is similar to the singular models because it provides an elaborate library of ready to use model structures of different models including HBV-96 and thus requires little effort in setting up an HBV variant. As for Raven, fluxes are elaborately described and linked to a singular model and the most important template input files are provided for six existing models. This includes the HBV-EC variant which differs from HBV-6 as it solves the energy balance and includes soil texture. Thus, this variant provides mostly building blocks from which the HBV-6 variant had to be assembled. SuperflexPy is more like a hydrological model programming language which helps model development by providing model elements as building blocks, though some building blocks of HBV-96 are provided as an example. Therefore, the available model components of SuperflexPy and Raven are not seen as a ready-to-use model and only the constructed variant is shown. This includes a level of subjectivity, therefore differences in model structure cannot solely be attributed to the original model but are the result of choices that every modeler will have to make within these frameworks.

3. Analyses and Methodology

In this section the four distinct parts of this study are explained in detail (Figure 2). The first part of our analysis explores the model structures. The goal is to obtain an overview of the differences, both mathematically and numerically, between the variants and the benchmark model in order to explain model output differences in the other three parts of the analysis. The second part compares model results across different levels of model complexity to assess the impact of numerical implementation differences on model output. The structured approach helps to identify which components cause the main model differences, but does not resemble model use in practice. In the third part, we evaluate the impact of mathematical model difference by switching-on model components within HBV-light (resembling “conscious” usage) and all intricate components that occur in other HBV-variants (resembling “off-the-shelf”). The last part corresponds with common modeling practice in which real forcing data and calibration is applied. All models variants are called from a set of R scripts (but run in their original programming language), in order to use the same parameter settings, sampling and calibration algorithm, and analysis criteria across all model variants. It should be stressed that all results are evaluated with respect to HBV-light as benchmark. Therefore, the analysis does not include evaluation of how well the models fit to observed data, but merely compares the model's ability to mimic model structure and reproduce modeled streamflow of an original, previously developed version (represented by HBV-6-light).

3.1. Model Structure Comparison

In the first step, we compare the included processes, the applied equations, and the numerical scheme of all HBV variants to the HBV-light benchmark. All models are HBV-based, but some variation in model structure might exist. Those differences can be used to explain model output differences. Besides, a detailed understanding of the model structure is needed to conduct the next part of this study, where we keep model structure the same to explore numerical differences. In this step, structure differences are categorized as either numerical or mathematical differences.

Gathering model structure information is not straightforward because, generally, not every paper provides a detailed numerical and mathematical description, and some details may be missing (Clark & Kavetski, 2010, as they also noted). Therefore, code was examined and a visualization of model results was used to infer information on the model structure. When a numerical description was found incomplete and/or code was inaccessible, model results were recalculated outside the model. This was done by using a trial and error method by taking one time step as the initial condition and comparing the modeled result to the results at the next time step derived from several numerical techniques. In addition, we contacted most of the model developers to verify our findings.

Mathematical model differences are evaluated with respect to their handling of the different hydrologic flux components within HBV. The fluxes are classified into the following groups to show similarity or type of difference compared to HBV-light:

- Identical (I) no difference between the HBV-light formulation and the variant formulation
- Complex - off (Co), the equation is more complex than the formulation of HBV-light, because a parameter was added. The complexity can be switched off, making it identical under certain conditions, for example, the extra parameter $q = k \times S^a$, $a = 1$ for linear outflow
- Simplified (S), the formulation is the same but a multiplication parameter is missing, for example, no snow correction factor
- Missing (M) the flux does not exist in this model variant
- Additional - off (Ao), the model has an additional flux that can be switched off
- Different (Db/s), the same process is represented differently. (-b/s) big/small the different part can be a major (b) or a minor (s) difference. Generally, big differences involves different parameter and small differences indicates a different restriction in the mathematical formulation, e.g., percolation is calculated as a linear function of storage instead of a constant value (Db) or snowfall = 0 instead of 1 for $T = t_t$ (Ds)

Some models have multiple classifications for one model component, because multiple parameters can differ in one formulation.

3.2. Impact of Numerical Implementation

Models that have the same numerical implementation and the same mathematical model should produce the exact same output. When the numerical implementation differs, but the included relevant processes are mathematically the same, the output should ideally be close to identical. In this part, the impact of numerical implementation differences on model output is evaluated, representing neatly mimicked models.

In order to isolate effects of numerical implementation differences from mathematical model differences, mathematical models that differ are excluded from comparison as soon as the differing model component is switched on, thus reducing the number of model variants for comparison. Switching a model component off/on is done by setting a parameter value (for all variants) such that it renders a flux to zero. Though a simpler model structure is good at explaining model output differences, such a structure does not normally represent reality well, as the model is too simple to catch the more complex system. To balance number of included models and model structure realism, the impact of the numerical implementation is assessed across increasing model complexity. This complexity is built up in 11 steps (configurations) by switching on model components. First, all model components are switched-off except for one linear outflow, such that the first configuration starts with a simple (leaky) bucket model which can be described by only two equations: $\frac{\partial S}{\partial t} = P - Q$ with $Q = k_1 \times S$ being the outflow. Then, step by step, extra model components are

switched on until the full model is run except for snow processes (a synthetic temperature time series is used to ensure that $T(t)$ is always higher than the threshold value for snowfall t_i). This procedure is repeated, this time including snow by allowing temperatures to drop below zero (snow processes include multiple mathematical model differences). Given that models are excluded as soon as their mathematical implementation deviates from HBV-light, 9 models are included in the first configuration with only a simple bucket switched on, and only 3 (MARRMoT2, EDU2, and Raven2) variants are left for the last configuration with all processes activated.

The order of switched on model components was chosen such that the maximum number of models are included for as long as possible (order shown graphically in Figures D2 and D3, other orders are possible by swapping the activation of certain processes while keeping the total number of included variants the same). In addition to the order of switching on processes, the number of models included can be increased by choosing the right parameter values and forcing data (described in the next paragraph), thus switching-off unnecessarily complex formulations (Co). For instance, a non-linear reservoir can have an exponent of 1, making it similar to a linear reservoir, and a constant temperature of 10°C switches snow processes off.

Synthetic forcing data is slightly different for the 11 configurations because it is in some cases used to switch model components off. It is created such that it is as simple as possible while addressing all model processes. This has the downside that it is less realistic because the link between temperature and evaporation misses. Precipitation events with different intensity and duration are alternated by recession periods to reset the models (all storages empty). In this manner, model response to a variety of precipitation events is tested. A spin-up and recession period of 400 days is used. A full forcing data description can be found in Appendix E. Parameter values have been selected to either be in the middle of the parameter ranges as suggested by Seibert and Vis (2012), or selected in such a way that it would switch off a certain process. Since there are no mathematical differences between the (simplified) models, the same parameter values could be employed for all variants. An overview of all parameter values can be found in Table A1. Similarity between the model variant hydrographs are evaluated against the HBV-light benchmark using the Kling-Gupta efficiency (KGE) metric (Gupta, Kling, et al., 2009) for the different (switched on) model structures.

3.3. Impact of Mathematical Model

In the numerical comparison, we looked at the effect of numerical differences on model output. We did, however, not cover the full numerical implementation differences of all variants because some variants are excluded as soon as their mathematical implementation started to deviate from HBV-light. Besides, only one value for each parameter was used and output differences might change for different parameter combinations. Both the parameter range as well as the full numerical implementation for all variants are included in this third part of the analysis. Furthermore, we allow for mathematical model differences. In this part, we assess model mimicry over a parameter range and for the full (activated) model structure. One set of 50 parameter samples is applied twice for all model variants (including HBV-light); once to represent "conscious" model use as explained in the next paragraph, and once to represent "off-the-shelf" model use. Another set of 50 parameter samples is only used for HBV-light to generate independent results.

The model result form an ensemble of 50 hydrographs for each variant. For each of those 50 hydrographs, the parameter set is changed. Seventeen out of the 24 parameter values that can be found across the model variants are varied (not all 24 parameters are found in all models, models contain between 14 and 19 parameters). These 17 parameters switch on model components of HBV-light (14 parameters) and competing mathematical formulations (a_{thorn}, per_{EDU}, and $alpha$). The three competing formulation parameters are needed for some variants to include the same processes even when it is mathematically different. The seven remaining parameters represent components in HBV variants that only add complexity compared to HBV-light and those components are switched off (fixed switch off value). In this way, mathematical model structure difference between the model variants and HBV-light is limited to a minimum, representing a "conscious" model comparison. The second application uses again 50 samples, but this time all 24 parameters are varied, even those that add complexity (possibly changing the perceptual model). This application represents a more "off-the-shelf" model use.

Parameters are sampled using an optimum Latin Hypercube Sample (McKay, 1979, lhs development) (Carnell, 2020, implementation in R). Both 50 samples were taken assuming a uniform probability distribution within the parameter range. The parameter range was the same for both samples and is shown in Tables A2 and A3. The same synthetic forcing data as for the last configuration of the numerical comparison (with snow process and evaporation on) is used.

The simulated streamflow differences are compared by calculating the KGE for each of the 50 ensemble members as evaluated against the results of HBV-light using the same parameter set (thus, the data is dependent for this KGE calculation). In addition to the KGE, the model ensemble of each variant is compared to the ensemble of HBV-light for each time step to differentiate between different parts of the hydrograph. Overlap of the ensembles is assessed with the non-parametric Mann-Whitney U-test (MWU) which tests if two samples come from the same distribution by calculating a shift in the mean (Bauer, 1972). This test is chosen because the results are often non-normal in the recession period, many tied values occur because of rounding in HBV-light, and the sample size is different for one variant (EDU, see Results section). The test is corrected for tied values using the method of Siegel and Castellan (1981). A separate parameter sample set was used to generate the ensemble of HBV-light, to ensure independent results from the model variant ensemble. This is a requirement for the MWU test.

3.4. Parameter Calibration

The previous section was limited to a sample of 50 parameter combinations. In "everyday model use", calibration is often used to find the optimal combination of parameters. With calibration, the parameters might also be tuned in such a way that they can compensate differences in model structure. This last analysis represents "everyday model use," where we employ real forcing data, and all model variants are calibrated on the simulated streamflow of HBV-light.

We employ the forcing of "HBV-land" (Seibert & Vis, 2012), which is based on the Swedisch catchment of the Svartån river at the Åkesta Kvarn station within the NOPEX (NOrthern hemisphere climate Processes land-surface EXperiment) experiment (Seibert, 1997; Seibert & Vis, 2012; Xu et al., 1996). The monthly potential evaporation is converted to daily potential evaporation with $E_p(t) = (1 + c_{et} * (T_{ave}(m) - T(t)) * E_{ave}(m)$ in which c_{et} is a parameter and T_{ave} and E_{ave} are monthly averaged temperature and evaporation respectively (similar to internal conversion in HBV-light, without linear interpolation). This method ensures that monthly or daily handling of evaporation input data do not cause any output differences between the model variants (Breuer et al., 2009, showed that this could cause substantial differences). A warm-up period of 0.67 years (243 days), calibration period of 6 years and evaluation period of 4.3 years (1,582 days) is used, similar to the suggested intervals in the HBV-land exercises as described in Seibert and Vis (2012). A single HBV-light simulation was used to generate the "observed" data which the variants are calibrated on. A Genetic Algorithm (GA) in R as described by (Scrucca, 2013) is chosen as calibration algorithm because it is similar to the built-in GAP optimization of HBV-light and could be applied to all model variants. The population size is 50 and 10 iterations are used ($50 \times 10 = 500$ runs per variant), other settings follow the default.

The calculation budget is for practical reasons (computationally inefficient connection between programming languages) one to two magnitudes smaller than common GA applications. Hence, 11 seeds of HBV-light are included in the calibration (next to the other models) to test the algorithm's ability to converge within the limited budget. The same 50×10 calibration budget is applied to every seed with the same parameter ranges except for the initial parameter set. As the observed data was created with the same model, a perfect solution exists which would result in a KGE of 1. Thus, the calibration results of HBV-light show to what extent the calibration is able to find the optimal solution. The convergence was found to be sufficient (presented in Result section) when applying an initial guess, parameter prioritization, and narrowing down the parameter range to accelerate calibration convergence (all explained in detail in Appendix F). Parameter prioritization allows for a limited number of varying parameters only, making it comparable to the "conscious" configuration in the previous part. Similar to the previous parts, the KGE is used as objective criterion to calibrate on and analyze the results. Modeled streamflow similarity of both calibration and evaluation data are assessed.

Table 2
Mathematical Model Comparison to HBV-Light

	Snow routine			Soil moisture routine				Groundwater routine				Routing routine
	P_s/P_r	M	R_{fr}	E_x/I	R_e	E_p	E_a	P_e	q_1	q_2	q_0	R_{ouq}
MAC	Db/Co	Db	M	M	I	Db	Ds	I	Co	I	I	I
TUW	Ds/Co	Co	Db	M	I	I	Ds	I	I	I	I	Co
EDU1	S/Ds	I	M	M	I	Db	Ds	Db	I	I	I	M
EDU2	I	I	I	I	I	I	I	I	I	I	I	I
MARRMoT1	Ds/S/Co	Co	Co	I	I	I	Ds	Ao	Co	I	M	I
MARRMoT2	Co	Co	Co	I	I	I	I	Ao	I	I	I	I
Raven1	Co	Co	Co	Co	Co	I	Ds/Co	Ao	Co	I	I	I
Raven2	Co	Co	Co	I	Co	I	I	Ao	Co	I	I	I
SuperflexPy	I	I	I	I	I	I	I	Ao	Co	I	I	Ds

Note. The results are classified as: I = identical, Co = complex - off, S = simplified, M = missing, A(o) = added(off), Db/Ds = different, b = big/s = small with corresponding colors showing magnitude of deviation. A graphical overview is presented in Figure H2. Flux formulation is given in Figure 3.

4. Results

4.1. Model Structure Comparison

First, we explored to what extent the HBV variants differed from HBV-light in terms of process formulation and numerical implementation. Numerical differences are shown in the last three columns of Table 1. The model description was missing for some models and did not always cover the entire numerical structure for others. Therefore, most of the results are based on code and trial-and-error calculations to reproduce the model output for several time steps, although this technique could not be applied when the state was not given as output. All models have a fixed time-stepping scheme and apply a first order numerical approximation. The majority of the models have explicit schemes, two have an implicit scheme, and TUW is partly based on analytical solutions. Most models solve their equations sequential, and only two models apply simultaneous solving. In SuperflexPy, each model component is solved sequentially but multiple fluxes can be calculated simultaneously within a model element. For this reason, the interdependent S_1 (snow) and S_2 (water in snow) had to be built into one model element.

Furthermore, some variants became unstable or showed unrealistic behavior (negative fluxes/storages, oscillation) for particular parameter settings. An overview of (numerical) adaptations is shown in Table B1, Appendix B presents other minor numerical implementation differences that were found.

The mathematical model comparison is shown in Table 2 and a graphical overview is given in Figure H2. None of the original model variants have exactly the same mathematical model as HBV-light. Even within Raven and MARRMoT, where many different formulations of the same process are available, the specific formulation for HBV-light was unavailable and new model processes had to be built to be able to create an identical mathematical model. Model code and the model description in the paper did not correspond com-

Table 3
Model Source Location and Used Version

HBV-light and HBV-land	4.0.0.23	https://www.geo.uzh.ch/en/units/h2k/Services/HBV-Model/HBV-Download.html
MAC	1.0.0	https://tuspace.ca/~rmetcalfe/MACHBV.html
TUW	0.1–8	https://github.com/cran/TUWmodel/blob/master/src/hbvmodel.f
EDU	1.0.0.0	http://amir.eng.uci.edu/software.php
SuperflexPy	0.2.2	https://github.com/dalmo1991/superflexPy/
Raven	2.9.2	http//raven.uwaterloo.ca/Downloads.html
MARRMoT	1.2	https://github.com/wknoben/MARRMoT

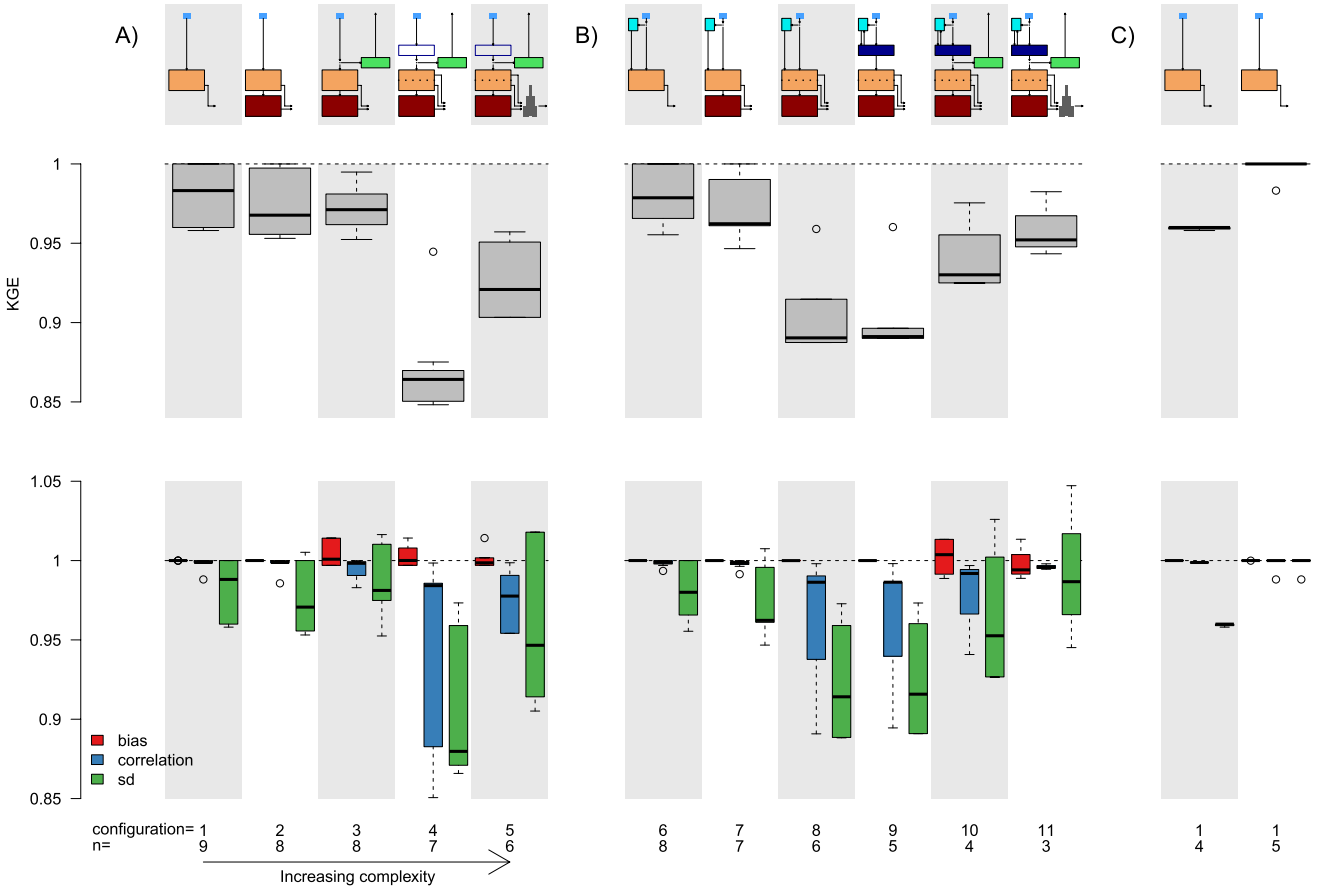


Figure 4. Impact of numerical implementation on model performance (compared to the benchmark HBV-light). The mathematical model is the same for each boxplot. The model structure complexity increases along the *x*-axis for panel A (configuration 1–5, without snow) and B (configuration 6–11, with snow). The number of included models (*n*) decreases as model components are switched on (indicated at the bottom). The switched on model components are illustrated at the top of each boxplot, following Figure 3. Model performance is expressed by the Kling-Gupta efficiency (KGE). The bottom panels split the KGE into bias, correlation and relative standard deviation. One outlier for the correlation in configuration 5 is not shown and outliers for the KGE in configuration 3, 4, and 5 are not shown. These outliers were caused by instabilities in MAC. Panel C shows the configuration 1 split up into implicit (left) and explicit (right) model variants. Both have the same model components switched on. TUW solves analytically, which is very close to implicit results and is included in the implicit boxplot.

pletely for certain processes for three different models. Additionally, the model structure needed to be tested for some specific situations because they were not (fully) described in their corresponding paper. Examples for snow threshold differences and evaporation restriction are elaborately described in Appendix C.

In summary, none of the model variants had originally the same numerical or mathematical model structure as HBV-light. The numerical structure could be determined for almost all models based on description, code, and testing. Some minor inconsistencies between description and code were found as well as instabilities for specific situations. Mathematical models were generally better described in the paper than numerical implementations, but specific situations still had to be tested. Besides different mathematical process formulation, fluxes were added and/or removed in most variants compared to HBV-light. Models that bear (part of) the same name or are inspired by the same model, can thus both have differences in numerical implementation and process formulation. In the next section we explore the consequences for model output.

4.2. Impact of Numerical Implementation Using Increasing Model Complexity

Figure 4 compares the impact of numerical implementation differences on simulated streamflow. All included models have an identical mathematical model structure at each level of increasing model complexity, hence, differences can only occur because of a different numerical implementation. Numerical

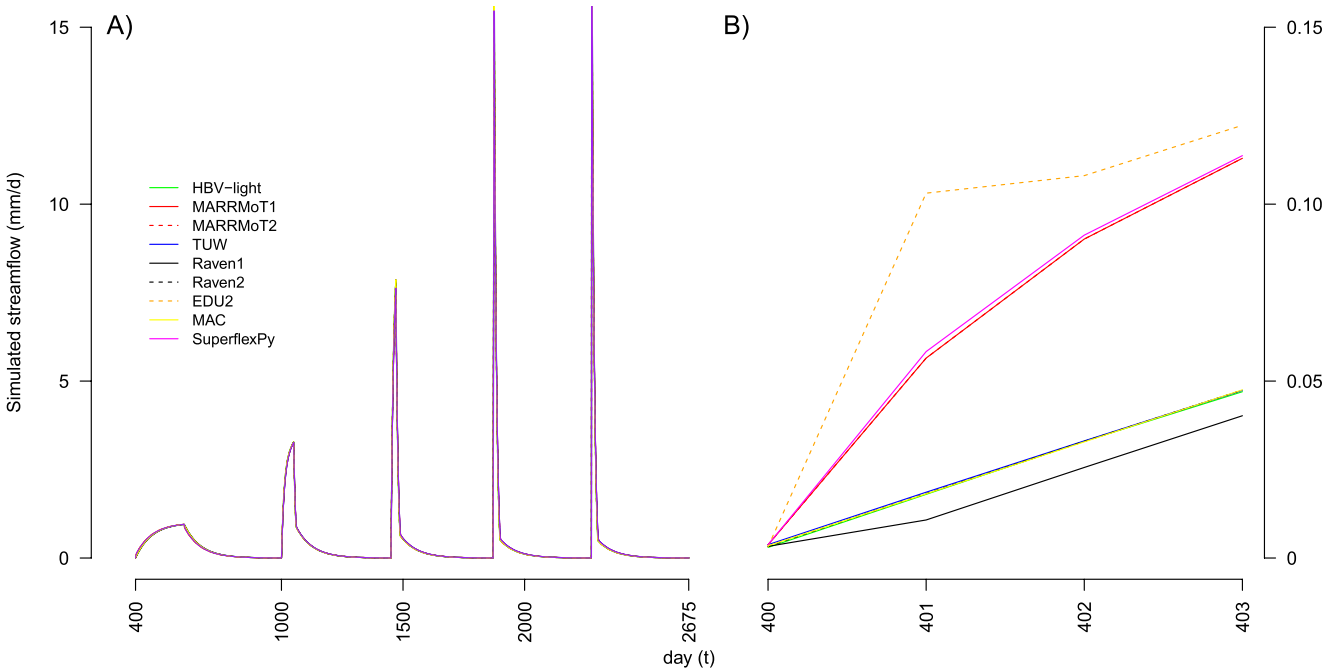


Figure 5. Impact of numerical implementation on simulated hydrographs for a simple bucket model with only percolation switched on. Panel A shows the whole hydrograph (without spin-up) and Panel B zooms in to $t = 400–403$. Impact of numerical implementation on simulated hydrographs. The results shown are differences for a simple bucket model with only percolation switched on. Panel A shows the whole hydrograph (without spin-up) and Panel B zooms in to $t = 400–403$. EDU1 is not shown because the mathematical model differs. The whole hydrograph shows barely observable differences with most lines overlapping (4th and 5th peak slightly higher than axis). An extensive zoom shows differences caused by the sequence order better.

implementation differences could cause a difference in the KGE metric of up to 0.15 (median up to 0.14) even when unstable model results are ignored. Switching on model components and increasing model complexity resulted in both an increase and decrease of output similarity. Routing had a clear positive effect (configuration 5 and 11 in Panel A and B, respectively, of Figure 4) because it smooths differences. The snowfall and melting process alone did not have a large impact on the KGE-value (comparing configuration 1 and 6), but fewer models were included (due to a different mathematical model) which could flatten the KGE change. A major KGE decrease (caused by the bias and standard deviation) is observed when quick flow is switched on, which could be explained by order of solving and to a lesser extent implicit/explicit scheme (comparing configuration 3 to 4 and 7 to 8). A minor decrease is noticed when the second reservoir is switched on (comparing configuration 1 to 2 and 6 to 7). The evaporation also has a major impact but this is better visible in the KGE scores of individual models (Appendix I), because models using an explicit Euler numerical scheme have a tendency to show lower KGE values which is compensated by higher KGE scores from models with implicit schemes in the aggregated KGE scores.

The differences between the variants can be traced back to the numerical implementation. This process is demonstrated for Figure 5 which shows the initial hydrograph response on rain when percolation is switched on. EDU1 is not shown because the mathematical model differs. The whole hydrograph shows barely observable differences with most lines overlapping (4th and 5th peak slightly higher than axis). An extensive zoom shows differences caused by the sequence order better. EDU2 calculates the percolation based on the previous time-step but outflow on the current time-step. Thus, there is only outflow and no percolation in the first time-step of the precipitation event. MARRMoT and SuperflexPy divide the precipitation over percolation and outflow while Raven, TUW, MAC, and HBV-light sequentially calculate the percolation flux before the outflow. Raven1 has the slowest response because of averaged flow. The difference between SuperflexPy and MARRMoT(1/2) is caused by a small excess storage in MARRMoT. TUW and MAC have a slightly higher initial position that explains the (barely observable) difference with HBV-light and Raven2 (all plotted on top of each other).

Assessing the impact of numerical difference more generally, implicit versus explicit numerical schemes has the most impact on simulated outflow. This was quantified for the first configuration by splitting up the model variants, shown in Figure 4c. The explicit models are clearly closer to HBV-light, which also employs an explicit numerical scheme. Furthermore, the order at which equations are solved has a major impact on model results, as shown for percolation in Figure 5. Besides percolation, the sequence also affects quick flow activation and how precipitation is divided over reservoirs S_3 and S_4. The implicit schemes play a role here as well. Numerical differences in both quick flow and evaporation cause HBV-light to have higher peaks than the other variants. This causes the relative standard deviation component of the KGE to be lower for most configurations. Other minor numerical differences that were visible in some of the hydrographs (not shown) are rounding, logistic smoothing, instantaneous flow and instability. Small differences that are not visible in the hydrograph, but might explain small differences between SuperflexPy and MARRMoT in KGE values, are solving algorithm, and error tolerance.

The differences between the variants can be traced back to the numerical implementation, with implicit versus explicit numerical schemes having the most impact. The first configuration is split up into explicit and implicit model variants, shown in Figure 4c. The explicit models are clearly closer to HBV-light, which also employs an explicit numerical scheme. Furthermore, the order at which equations are solved has a major impact on model results. Figure 5 shows the initial hydrograph response on rain when percolation is switched on. Differences are explained in the figure caption. Besides percolation, the sequence also affects quick flow activation and how precipitation is divided over reservoirs S_3 and S_4. The implicit schemes play a role here as well. Numerical differences in both quick flow and evaporation cause HBV-light to have higher peaks than the other variants (i.e., other [stable] variant maximums are 9%–27% with an average of 20% lower for the highest peak in configuration 4, with quick flow and evaporation switched on). This causes the relative standard deviation component of the KGE to be lower for most runs. Other minor numerical differences that were visible in some of the hydrographs (not shown) are rounding, logistic smoothing, instantaneous flow and instability. Small differences that are not visible in the hydrograph, but might explain small differences between SuperflexPy and MARRMoT in KGE values, are solving algorithm, and error tolerance.

Numerical implementation differences led to differences in model output with implicit and simultaneous schemes causing the largest differences in KGE (with HBV-light using an explicit and sequential numerical scheme, analysis not shown for brevity). Smaller KGE differences were also observed and could be related to other differences in numerical implementations such as rounding, etc. (as mentioned earlier in this section). All models were included when all components were switched-off, leading to close to identical output for explicit variants. Switching-on components resulted in both more and less similar model output. More similar output was mostly caused by the averaging out of multiple differences (e.g., routing or switching on off evaporation, configuration 4 to 5 and 2 to 3). A KGE metric difference of up to 0.15 KGE and a simulated streamflow peak reduction of up to 27% were observed, solely caused by different numerical implementation.

4.3. Impact of Mathematical Model Using Parameter Sampling

For the third part of this study, we look into the combined modeled streamflow difference caused by numerical and mathematical differences. The results of 50 parameter sample sets for "conscious" sampling (mathematical model difference are kept to a minimum) are shown in Figures 6a and 6b. For each parameter sample set, a KGE is calculated with the simulated streamflow. Allowing for mathematical model differences resulted in lower model mimicry with an average KGE difference of 0.27 (for comparison; a maximum difference of 0.15 for numerics only in the previous section). MAC and EDU1 mimic worse than they would mimic for a realistic forcing case, because of the missing link between evaporation and temperature in the synthetic forcing. This causes their alternative evaporation formulations to be more different than in realistic situations.

Raven2, EDU2, and MARRMoT2 are mathematical identical to HBV-light and are ranked 1, 2, and 4 when comparing the highest median KGE (0.97, 0.96, and 0.93 respectively; SuperflexPy has a median KGE of 0.94) and the variants that have more mathematical differences mimic worse. The difference between Raven2 and MARRMoT2 (0.04 KGE median) and the difference between MARRMoT2 and SuperflexPy (0.01 KGE me-

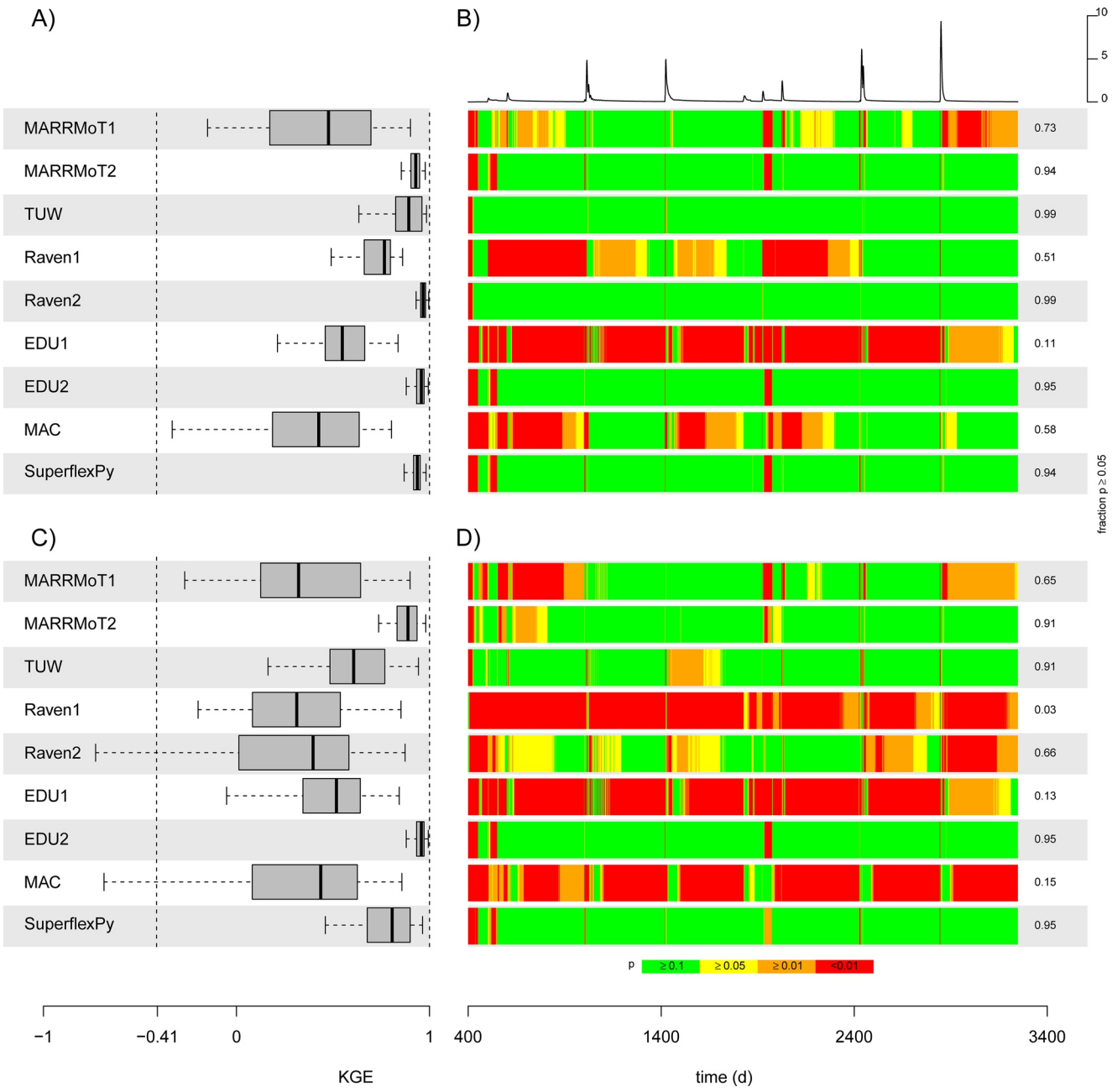


Figure 6. Impact of mathematical model on modeled streamflow differences for "conscious" (panel A and B) and "off-the-shelf" (panel C and D) parameter sampling. For the top windows, 50 parameters sets of all HBV-light parameters (14) and 3 parameters that substitute competing processes are used whereas all parameters that occur in HBV-variants are sampled in the bottom windows (14 HBV-light and 10 other parameters). For the left panels (A and C), each individual sample is compared with each individual sample of HBV-light with Kling-Gupta efficiency (KGE), whereas the right panels (B and D) tests overlap of the model ensemble for each time step (Mann-Withney-U). Outliers are not shown. The dotted line at −0.41 indicates 0 predictive value compared to the mean value of the simulated discharge of HBV-light (Knoben, Freer, & Woods, 2019). The number on the right indicate the fraction of the time that $p \geq 0.05$ (so the fraction of time that the output from the model variant and the output from HBV-light appear to be drawn from the same distribution).

dian) illustrate that mathematical model differences can be as big as numerical implementation differences. Magnitude of differences in modeled streamflow are not only dependent on the number but also the type of mathematical differences. TUW and MARRMoT1 have four and Raven1 has one different formulation, but when ranked on KGE median, Raven1 performs in between and the difference between TUW and MARRMoT1 is quite large (0.48, 0.77, and 0.89 KGE median for MARRMoT1, Raven1, and TUW respectively). This difference becomes even more apparent when looking at the MWU-test results (see paragraph below).

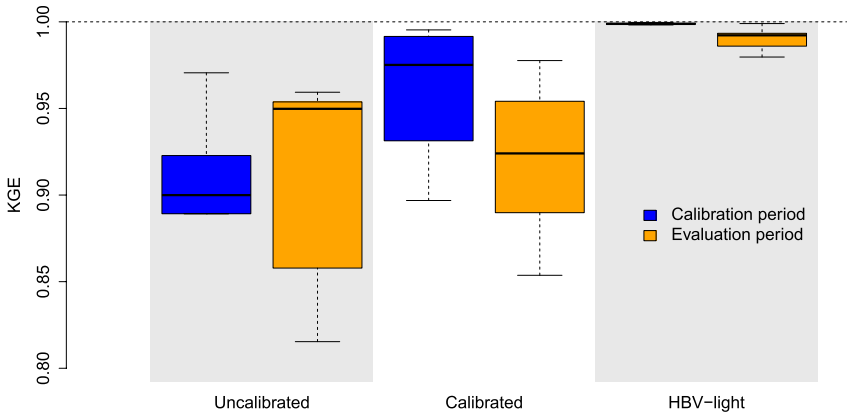


Figure 7. Distribution of model performance for calibrated model variants. KGE (Kling-Gupta efficiency) is determined based on model variant output against HBV-light, using real forcing data. Uncalibrated results use the same parameter set that was used to generate the solution with HBV-light. The KGE is calculated for each model variant and all variants together are shown as boxplot. MARRMoT1 is an outlier in the two uncalibrated boxplots and falls outside the plotting area. The genetic algorithm is used to calibrate the four most sensitive parameters with a population of 50 and 10 iterations. The third and fourth boxplot show the KGE for the calibrated models. The parameter settings in HBV-light were used to generate the "observed" data. Those settings were given to one of the individuals of the first generation within the population for all variants to increase convergence speed. The last two boxplots show 10 calibrated HBV-light samples with the same settings. The last column shows the algorithms capability to converge to an optimum given the limited calibration budget.

The overlap of the distribution can be tested for each time step. This is done with a Mann-Whitney U-test (MWU), shown in Figure 6b. There are some differences compared to the KGE distribution. It shows that the peaks are always poorly mimicked (visible as small red lines) in the model at $t = 502$, 1,003, 1,422, 1,932, 2,427, and 2,845. HBV-light has multiple 0 values from $t = 400$ onward due to rounding which explains the red area at the beginning of all model variants. For the MWU values, the peaks become less important and models that over- or underestimate those peaks, but have small errors otherwise (TUW, MAC), get higher MWU values than models that are always close to but never exactly the same as HBV-light (SuperflexPy, EDU1).

Continuing on the example of the previous paragraph, Raven1 drops in the ranking of fraction of the time MWU value is greater or equal than 0.05 (MWU values of 0.51, 0.73, and 0.99 MWU value for Raven1, MARRMoT1, and TUW respectively). The gap between the well mimicking models and poorly mimicking models is also more distinct. Furthermore, the specific time steps in which numerical implementation differences results in simulated outflow differences become visible. The variants generally poorly mimic the precipitation peaks with moments where $p < 0.01$ for all variants. However, some do better than others. For example, the red zones at $t = 1{,}850$ are mainly visible for the models that solve simultaneously, in which more water is routed to S_4 and the hydrograph rises quicker, than the sequential solving models, that fill up S_3 first and thereafter route most water to the slow responding lower storage (S_5). Another example is the beginning of the series. Significant differences ($p < 0.01$) between the variants and HBV-light are caused by rounding in HBV-light and the explicit infiltration splitter which routes most water flow to S_3 (soil moisture) whereas other models route more water to S_4 (upper soil zone).

There are multiple parameters in model variants that add complexity that are not needed when mimicking HBV-light. In the second sampling configuration, the same parameters were sampled, but this time also the extra parameters, introduced in the model variants, were included in the sampling (partly including perceptual model differences). Thus, the full extent of mathematical model differences become visible. The KGE distribution and MWU values are shown in Figure 6d. The average KGE metric dropped from 0.77 to 0.51 and the variants medians range changed from 0.43–0.97 to 0.31–0.96. The effect of specific mathematical components can be analyzed. Table A3 shows the model components that belong to each model. Most of those were not switched on in the previous sampling. For example, MARRMoT1 switched on the snow/rain interval, capillary rise and non-linear outflow, TUW switched on snow-rain interval, melt temperature, and croute (an alternative routing parameter). When comparing Figures 6a–6c, MARRMoT1 decreases not as

much in the KGE value as TUW. Extra model components have different effects, but they generally cause a reduction in the model variants' ability to mimick HBV-light. Thus, models with many extra components and options (Raven) are mostly decreasing in their KGE score for mimicry. The MWU-test shows the same decrease but the effect on the *p*-value is less strong. For example, Raven1 and Raven2 are still very differently colored (0.03 and 0.66 MWU value) while the boxplots are close to each other (KGE medians of 0.31 and 0.40). Further, SuperflexPy has a higher MWU value (0.94–0.95) while the KGE median decreased (0.94–0.81 comparing Figures 6b–6d).

In summary, mathematical model differences led to major output differences, especially when also the parameters were sampled from processes and options that were added in the model variants compared to HBV-light (KGE medians between 0.31 and 0.97). Model variants were quite different in output similarity with mathematically more similar variants having more similar output. Numerical implementation difference was still observable but often smaller than mathematical model differences. The discharge peak magnitude of the model variants was always smaller than HBV-light due to a different numerical implementation.

4.4. Parameter Calibration

In this last section, we explore model mimicry in an "every-day model use" setting. All model variants are calibrated on the simulated streamflow of HBV-light. The calibration results are shown in Figure 7. The non-calibrated models variants differ from HBV-light, but mimic a lot better than in the sampling exercise of the previous part due to the real forcing data (average KGE of 0.77–0.81 and median KGE of 0.73–0.91). MAC, Raven1, MARRMoT1, and EDU1, in particular, benefit from forcing data that includes less extreme precipitation events and a temperature-potential evaporation link that better mimics observations.

First, we tested if the calibration budget was sufficient to find the optimal parameter values. HBV-light was calibrated to its own data, which should ideally lead to a KGE of 1. The average KGE value of the 10 HBV-light calibration results was 0.9988 with a standard deviation of 0.0004. This demonstrates that the calculation budget is sufficient to get close to the optimal parameter values. Subsequently, the model variants were calibrated on HBV-light its simulated streamflow. When calibrated, the model variants increased from a median KGE of 0.90–0.98 for the calibration period, but none of the models could reproduce HBV-light well enough to come within two standard deviations of the HBV-light mean. This is emphasized by the differences in calibrated parameter values show in Table G1.

The effect of numerical implementation differences after calibration for the calibration period was tested separately as well. This was done by conducting a one-sided Welch-test, to test if the KGE mean of the mathematical identical models (EDU2, MARRMoT2, and Raven2) was lower than the KGE mean of the HBV-light samples. The numerical differences alone resulted in a lower KGE than the HBV-light samples, but the differences were insignificant with $p = 0.09$.

The evaluation period had lower KGEs than the calibration period for all three boxplots. HBV-light dropped to an average KGE of 0.990 with a standard deviation of 0.006. The uncalibrated models show mixed responses, but show an overall average KGE increase because of the increase of MARRMoT1 from 0.10 to 0.86. This increase is related to MARRMoT1's missing quick flow which results in poor peak similarity. The evaluation period had less extreme events with the seven highest discharge peaks all falling in the calibration period. When calibrated, most models increase slightly for the evaluation period. Only EDU2 lies within two standard deviations of HBV-light. For the evaluation period, numerical implementation differences resulted in a lower KGE value than the HBV-light samples after calibration with $p = 0.04$ for the one-side Welch-test. This proves that numerical differences alone have a significant impact on simulated streamflow, even after calibration.

In summary, modeled streamflow of mimicking models was more similar to HBV-light after calibration than with parameter sampling (KGE average of 0.77–0.91 and median of 0.73–0.97), but differences were still significant ($p < 0.05$). Calibration increased model output similarity for the calibration period (KGE median 0.90 to 0.98), but reduced for the evaluation period (KGE median 0.95 to 0.92). Real forcing data increased model output similarity compared to the synthetic forcing data (KGE average of 0.77–0.81). The mathematical identical variants, thus only differing in their numerical implementation, provided more similar but still significantly different results compared to HBV-light, even after calibration.

5. Discussion

We started with the rationale that it would be easier to study if and why similar models behave differently than why different models behave similarly. To this end, we compared model structure differences and simulated outflow differences of models that bare the same name or are inspired by the same model. It was expected that the model structure would be fairly similar; identical mathematical models produce close to identical output, and mimicking models provide similar results. We found that model structures differed on numerical, mathematical, and even perceptual level with none of the original variants having the same numerical or mathematical model as HBV-light. This can be because modelers might have reasons to deviate from their source model (in this case HBV), the source model may have different versions, or modelers are not completely able to copy the model structure because it is not based on (open-source) code but instead on, for example, model descriptions. The result is that models that bear the same name, can still produce substantially different model results (KGE average of 0.51 for "off-the-shelf" model-use, Part 3). Below, we discuss reasons for the model structure differences found in this study. Thereafter, we suggest ways forward to address these differences and improve model mimicry capacity.

5.1. Why do Mimicking Models Not Mimic Exactly?

We expected that numerical robustness, simplification and a conscious different representation of reality would be the main reason for model structure differences. This might be the case for some differences, which could be linked to the purpose of the model, for example, educational purposes, or data scarcity. However, three other less obvious reasons are noticed which could explain differences as well. The first explanation is the precision of a model structure description. For some models, details on mathematical model aspects were missing in the corresponding paper descriptions (elaborate example in Appendix C). We found, for instance, four different outcomes for the precipitation splitter into rain and snow when the temperature is exactly equal to the threshold temperature. Next to this, HBV-light was the only model which had an evaporation limitation based on snow cover. That this was not included in the mimicking models is not surprising, given that these restrictions are not described for HBV-6 nor for HBV-light (it is also not described for HBV-96, but implementation was not tested in this variant).

While one needed to search for the missing mathematical model aspects, an incomplete numerical implementation description was even more common. Although the relative impact of numerical differences compared to mathematical difference was generally smaller, it was still considerable and even dominant for peak streamflow (up to 27% lower peaks by numerical differences, although the application of real precipitation is expected to be smoother and reduce differences). Differences of this magnitude could be of importance for example for flood risk management. This importance is not reflected in the unequal attention that is given in the model description, where the mathematical model is described extensively while the numerical implementation is in some cases completely ignored. Although a limited number of models was used for a limited number of parameter combinations, the findings of Clark and Kavetski (2010) support the claim that numerical choices can have a major impact on model results, especially for extreme precipitations events (La Follette et al., 2021). Furthermore, they noticed a more widespread minimal numerical implementation description. Thus, incomplete numerical description seems to be a common problem for hydrological models.

Another cause is uniqueness of model formulation. Most characteristic of the HBV-model is the infiltration splitter which is clearly described and mathematically identical for all variants. However, less characteristic components of the model are more easily adapted. For instance, evaporation has six different representations in seven models and even within HBV-light two options are available. Thus, the name HBV seems more linked to one unique model component than to the whole structure. From this perspective, it is surprising to see the differences in snow process representation since HBV is originally built and most famous for its snow implementation (snow differences description in Appendix C).

A last explanation is historical development of a model. In Section 2 we indicated that the model variants are either based on the HBV-6 of the HBV-96 version while they are compared to the HBV-6 version (represented by HBV-light). The models variants were often found to have a model structure which is a mix between those two versions. This vague origin was also found in the referencing, where some of the papers related to the model variants explored in this study clearly indicate their source model, while others are

less explicit. Besides, the used HBV-light variant was assumed to correspond to HBV-6, but this assumption could not be tested and mathematical differences like the evaporation restriction for snow >0 are possibly deviating from HBV-6. Mixing of historical versions is not restricted to HBV only, but could reasonably be expected to occur for every model that has been published multiple times through ongoing development. The parent model could be made more explicit when a systematic model naming is standardized.

5.2. Reproducibility of Models

The problem of an incomplete model description is worsened when model code is close source or not accompanied with an elaborate explanation. For this study, HBV variants were selected based on their open source, but this does not necessarily mean that the code is accessible to all users since there could be an executable wrapped around the code, one could be unfamiliar with the programming language, or code explanation/manual could be limited. Even within this selection and after elaborate output comparison to similar models, the model structure was not always known, underlining the importance of a detailed description. Both the extent of model structure description and open source code could explain why models are so far rarely mimicked. Hutton et al. (2016) argue that close source code and data are one of the main reasons why results are rarely reproduced, affecting the very basis of scientific advancement (Melsen, Torfs, et al., 2017). Reproducibility is especially relevant for model mimicry in the context of MMFs, illustrated by the difference in similarity between the original and mimicked model in Raven and MARRMoT. Raven is based on the original model code and is almost identical, while MARRMoT is based on model description only, leading to considerable differences (Craig et al., 2020; Knoben, Freer, Fowler, et al., 2019). Knoben, Freer, Fowler, et al. (2019) also address reproducibility and argue that results based on only partly known model implementation limits the generalizability of findings. However, they also noticed that data sets are more often shared and journals start to enforce sharing practices. The trend towards more attention for reproducibility is also noticed in this study based on the fact that the more recently developed HBV variants describe the numerical implementation more extensively.

5.3. What's in a Name?

The link between model output and model structure is a key aspect of model mimicry. In order to mimic, some level of similarity between original and mimicked model is required. However, similarity on at least one level of the model development steps (Figure 1) is not found for all models bearing the same name, as shown for HBV in this study (although including the version reference, present for some variants, into account would give a slightly more nuanced overview). They differed not only at the numerical and mathematical level, but at the perceptual level as well with multiple fluxes and storages being added and removed. Despite the difference, some of the variants did not add a suffix to indicate the variation either. When the dominant processes are not even considered similar, which aspect justifies the use of a model name and what meaning is left in this name?

One reason to use a name could be to build on the legacy of an existing model (Addor & Melsen, 2019). From a modeler developer perspective, linking a newly developed model to a more famous, generally accepted model could help to communicate the idea behind the model. Furthermore, when building on a known model structure, a full model structure description and justification is not always demanded and only differences need to be described. Thus, using an existing model name could help to get a new idea across. However, different levels of similarity dilute the meaning of the name. Therefore, development of a systematic model naming framework is suggested for future model development research. Such a framework would link different levels of similarity to different levels of a model name. A classification system already exists for distributed models (Kampf & Burges, 2007), but a similar approach would be more difficult for conceptual/non-spatially distributed models which have a wider range of process representations. Knutti et al. (2013) proposes model genealogy expressed as family trees to show how climate models develop and why certain components are changed and others not. This view is based on an evolutionary view on model development which favors well working ideas and interchange those ideas with components from other models. Those dendrograms are closely related to the classification of Skidmore (2002), who argues that a taxonomic organization will help with model comparison and model selection for environmental models. A similar organization could be developed for hydrological model naming as well, at least at the model family

level, although challenges remain (Remmers et al., 2020). Model comparison could be supported by the recently suggested universal graphical model representation of Bancheri et al. (2019). This representation helps to identify differences in mathematical model structure easily. A hydrological model naming framework could be based on model comparison. This comparison is not expected to explain every idiosyncrasy at once, but rather starts with advancing general model behavior understanding. This way, improved understanding can highlight the added value of new models and helps conscious model selection. When done thoroughly, poorly described model structures of already existing models can be analyzed by comparison to similar models. A tool for systematic comparison could be the step-wise switching-on of model components as implemented in this study. It proved itself as a powerful tool to track down and explain model structure differences, especially when multiple variants are included. The comparison results could form a model family tree based on the levels of similarity. Ideally, a family tree would reflect both levels of similarity, like a dendrogram, and the relation to the model(s). The same levels of similarity could be expressed in levels in the name, linking numerical, mathematical, and perceptual model difference to other family members. For example, "HBV" is the model family, "-6″ the original version, "light" the mimicking model, and "-lumped-daily evaporation" the specific model structure within HBV-light. An important condition is that within a model, a name is given to all available model structures and model versions so differences are noticed and the correct version can be referenced when the model is mimicked.

Along with open source code and a more elaborate model description, a model naming framework creates a clear expectation of similarity between models and could help to appreciate and communicate new models and their relation to other variants. Thus, these three key aspects are ways forward to be able to reproduce previous results, build on those findings with mimicking models, understand output difference, and improve catchment understanding.

6. Conclusions

The idea of model mimicry is that a model structure is imitated. This allows for comparison between model components that represent different ideas about catchment functioning, an especially relevant aspect for modular modeling frameworks. We compared model structure and model output of seven models against the model by which all seven models were inspired and found that both numerical implementation and mathematical model were different for all models, thereby affecting modeled streamflow. The effect of multiple numerical differences could be recognized separately causing a difference in the KGE metric of up to 0.15 on a synthetic hydrograph. The combination of mathematical model and numerical implementation differences resulted in an average difference in the KGE metric of 0.27 in modeled streamflow for parameter samples. Numerical implementation differences were still observable in output (and sometimes dominant) but often smaller than mathematical model differences. Streamflow simulation differences decreased considerably after calibration of the model variants to the simulated streamflow of the benchmark, though this could partly be contributed to the use of more realistic forcing data. Calibration improved output mimicry for the calibration period (KGE median of 0.90–0.98 and average 0.80 to 0.90) but the effect on the validation period was less clear (KGE median of 0.95 to 0.92 and average 0.89 to 0.92). The mathematical identical variants, thus only differing in their numerical implementation, provided more similar but still significantly different results compared to the benchmark, even after calibration. To the best of our knowledge, this is the first study that identifies and links model structure and model output differences for both numerical implementation and mathematical model in such a rigorous method.

Several reasons for differences in model structure of mimicking models were discussed, with inaccessible model code and a limited numerical and mathematical model description being some of the main reasons, leading to a diverging concept of a model name.

We argue that the shared use of the name "HBV" by these models is not indicative of their internal behavior and can potentially be confusing. More systematic model naming is suggested to indicate the level of similarity between models. Such a model structure should include a reference to the model ancestor and have different levels in the name which reflect different levels of model similarity. Additionally, we propose rigorous comparison with other model family members when introducing new models, in order to advance future model mimicry.

Appendix A: Parameter Description and Values

Table A1
Description of the HBV-6 Parameters

Parameter	Process	Unit	Description
t_t	Precipitation/melt/refreeze	°C	Temperature threshold determining snow- or rainfall, and melt or refreezing
c_{fmax}	Melt/refreeze	$mm°C^{-1}d^{-1}$	Degree-day factor of snow melt and refreezing
s_{cf}	Snowfall	–	Snowfall correction factor
c_{wh}	Excess flow	–	Maximum water holding content of snow pack
c_{fr}	Refreeze	–	Coefficient of refreezing of melted snow
F_C	Iniltration/recharge	mm	Maximum soil moisture storage
L_P	Evaporation	–	Soil moisture value above which E_a reaches E_p (wilting point)
beta	Iniltration/recharge	–	Non-linearity coefficient of upper zone recharge
per	Percolation	mmd^{-1}	Maximum rate of percolation to S5 (lower zone)
u_{zl}	Quick flow	$mmmm°C^{-1}d^{-1}$	Threshold for quick flow
k_0	Quick flow	d^{-1}	Runoff coefficient for quick flow (from S4, upper zone)
k_1	Normal flow	d^{-1}	Runoff coefficient for normal flow (from S4, upper zone)
k_2	Base flow	d^{-1}	Runoff coefficient for base flow (from S5, lower zone)
max*bas*	Routing	D	Flow routing delay
t_{ti}	Precipitation	°C	Interval length of rain-snow spectrum
t_{tm}	Melt/refreeze	°C	Threshold temperature for snow melt refreezing
t_r	Precipitation	°C	Threshold temperature above all precipitation is rain
t_s	Precipitation	$mmmm°C^{-1}d^{-1}$	Threshold temperature below all precipitation is snow
alpha	Normal/quick flow	$mmmm°C^{-1}d^{-1}$	Non-linearity coefficient of runoff from S4 (upper zone)
c_{route}	Routing	$mmmm°C^{-1}d^{-1}$	Free scaling parameter for routing
r_{cr}	Rainfall	$mmmm°C^{-1}d^{-1}$	Rainfall correction factor
per_{EDU}	Percolation	$mmmm°C^{-1}d^{-1}$	Lineair coefficient for percolation to S5 (lower zone)
cap	Capillary rise	$mmmm°C^{-1}d^{-1}$	Maximum rate of capillary rise to S4 (upper zone)
a_{ihorn}	Evaporation	$mmmm°C^{-1}d^{-1}$	Coefficient of a simplified version of Thornthwaite formula
c_{et}	Evaporation	$mmmm°C^{-1}d^{-1}$	Correction factor determining evaporation based on long-term mean data

Note. Parameters below the horizontal line are parameters used by other HBV-variants.

Table A2
Overview of Parameter Settings for the Second, Third, and Fourth Part of the Study (see 2)

Parameter	t_t	c_{fmax}	s_{cf}	c_{wh}	c_{fr}	F_C	L_P	*beta*	*per*	u_{zl}	k_0	k_1	k_2	max*bas*
Sampling lower	−1.5	1	0.4	0	0	50	0.3	1	0	0	0.1	0.01	0.001	1
Sampling upper	2.5	10	1	0.2	0.1	100	1	6	3	70	0.5	0.4	0.01	7
Base value	0	5.5	0.7	0.1	0.05	75/*c*	0.65	3.5	1.5/*c*	35/*c*	0.3	0.1/*c*	0.015	4
Switched off	T (0.5)	0	1	0	0	(E)	(E)	(E)	0	(k0)	0	–	(per)	1

Note. The first two rows are the sampling range which is based on Seibert and Vis (2012), complemented by Uhlenbrook et al. (1999) when the former does not cover certain parameter ranges. The exceptions are the t_t range and k_2 upper value. The range of t_t is limited to the minimum temperature alternation and the k_2 value is restricted because MAC often crashes for $k_2 > 0.01$. The parameter values for the second and fourth part are set in the middle of the parameter range (logistic middle for the k_x values). A *c* indicates that this parameter is calibrated in the fourth part and that the sampling range is used initially. The calibration range was narrowed down for the final calibration. Switched off model values override the base value to simplify the model in the second part. T = temperature and E_p = evaporation.

Table A3
Overview of Settings of Parameters That are not in the HBV-Light Model

Parameter	t_{ti}	t_{tm}	t_r	t_s	*alpha*	c_{route}	r_{cr}	per_{EDU}	*cap*	a_{thorn}	c_{et}
Sampling lower	$t_r - t_s$	$t_t - 0.5$	t_t	$t_t - 2.5$	−0.5	0	0.5	*per*/60/0.1	*per*/5	0.1	0
Sampling upper	$t_r - t_s$	$t_t + 0.5$	$t_t + 1.5$	t_t	1	50	1.5	*per*/20/0.5	*per*/1	0.3	0.3
Base value	0	t_t	t_t	t_t	0/0*/*c*	0	1	0/*s*/*c*	0	0/*s*/c_{et}	0/0/0.15
Model using parameter	2, 4	2, 3, 4	3, 6	3	2.1, 4, 6	3	4, 6	5.1	2, 4	6	5.1

Note. The first two rows are the sampling range which is based on the paper that describes the parameter (see Table 1 for references). This sampling range is only used in part three, the second ("off-the-shelf") sampling. The base value is used in most configurations. Most base values switch the intricate parameter off except for per_{EDU}, a_thorn, and c_{et}. For those parameters, the first value is used for the second part (increasing model complexity), the second for the limited sampling, and the third value is the fixed value for calibration. The *s* indicates that the parameter is still sampled and the *c* indicates that this parameter is calibrated. *alpha* is sampled and calibrated only for MARRMoT1 because it misses a quick flow. The last row indicates which models uses which parameter with 2 = MARRMoT(0.1/2), 3 = TUW, 4 = RAVEN, 5.1 = EDU1, and 6 = MAC. Some of the parameters are coupled to another parameter because they represent the same process.

Appendix B: Changes to Original Model

B1. Minor Numerical Implementation Differences

Next to the implicit/explicit scheme and order of solving several minor numerical implementation differences were found which are described below. Several (numerical) implementations have been adapted. Those adaptations are shown in Table B1.

The solvers that were described are: a Newton solver in Raven, Pegasus (Dowell & Jarratt, 1972, advanced bisection) method in SuperflexPy, and MARRMoT uses the Matlab built-in solvers *fsolve* and switches to *lsqnonlin*, when a certain accuracy threshold is not met by *fsolve*. Some other minor numerical differences have been

Table B1
Changes That are Made to the Original Models

Model	Model part	Reason
MARRMoT2	Numerical, infiltration	Crashes
MARRMoT2	Excess	Negative flux, oscillation in S_2
MARRMoT2	Melt	Implicit melt function only halves when S_1 runs empty
MARRMoT2	Infiltration	Negative flux
MARRMoT2	Snowfall	Add s_{cf}, if $T = t_t$ $P_s = P$ $P_r = 0$
MARRMoT2	Actual evaporation	Restrict $E_p = 0$ for $S_1 > S_{tol}$
MARRMoT2	Normal and base flow	add q_0 and remove alpha
Raven1	Multiple- like GR4J emulation	HBV-EC is distributed and solves energy balance, many parts changed
Raven2	Evaporation	Restrict $E_p = 0$ for $S_1 > S_{tol}$
Raven2	Numerical	Instantaneous flow corresponds with HBV-light
EDU2	Numerical	Change infiltration flux, complex numbers
EDU2	Numerical	$S_3 = \max(F_C, S_3)$, less oscillation
EDU2	Snowfall	Add s_{cf} missing in original
EDU2	Routing	Routing added
EDU2	Potential evaporation	Average monthly E_p to daily E_p
EDU2	Actual evaporation	Restrict $E_p = 0$ for $S_1 > S_{tol}$
EDU2	Percolation	Based on storage instead of fixed
EDU2	Refreezing	Missing in original
EDU2	Excess	Missing in original
EDU2	Rainfall	$P_r = 0$ for $T = t_t$
SuperflexPy	E_a	Restrict $E_p = 0$ for $S_1 > S_{tol}$
SuperflexPy	Quick flow	Missing in original
TUW	Snowfall	$t_r = t_r + 1e{-}10$

observed in the model descriptions. MARRMoT has smoothing functions implemented for temperature and storage thresholds, which is not described for other models. MAC and HBV-light round their modeled streamflow (we used the default of 3 decimal places for HBV-light). Furthermore, Raven averages the outflow over the time step by default (this is switched-off in Raven2). Moreover, the restrictions also differ. For instance, negative flux values and storages were noticed for EDU (major) and MARRMoT (e−4) while Raven had a negligible negative storage (e−17), whereas this was not found in the output of the other models. Non-negative flux restrictions are described for some models but others needed manual testing. EDU and MAC became unstable for some parameter samples, leading to oscillating and unrealistic results (complex numbers, no discharge response to precipitation). Numerical changes were made to EDU2 to avoid crashing and most of the unrealistic behavior. TUW originally produced NaN results for $t_r = t_s$ (temperature thresholds for dividing precipitation over rain and snowfall) which was overcome by a small model adaptation (Table B1). MARRMoT1 showed major oscillation in the S_2 storage for the rising limb of more intense precipitation events (before numerical change) and MARRMoT2 required a numerical adaptation in the infiltration flux to avoid crashing.

Appendix C: Incomplete Model Restriction Examples

The mathematical model comparison was show in Table 2. To come to this overview, model description was studied. Model code and description did, however, not always match and therefore they needed to be tested. Below, two examples of incomplete model description are described.

For the snow/rain threshold, most papers describe what happens when 1 mm of precipitation falls for $T > t_t$ or $T < t_t$ but not for $T = t_t$ (where T = temperature and t_t = snow/rain threshold). When $T = t_t$ is combined with switching the snow/rain interval off ($t_{ti} = 0$ for the models that include the interval), HBV-light has 100% of snow and 0% rain (of the precipitation); EDU and MAC have 100% of rain and 0% of snow; TUW originally provides NaN and with the adaptation it has 100% of snow and 0% rain, and MARRMoT1 has 0% of both rain and snow (although using the discrete snow/rain threshold formulation instead of an interval results in 50% of both snow and rain).

Another example is that the actual evaporation equals 0 when snow >0. Only HBV-light has this restriction implemented and TUW reproduces this as actual evaporation = 0 for $T < 0$. For both model variants, this restriction is not described in the corresponding paper. This information could only be retrieved by looking closely at the model output of the numerical implementation comparison.

Appendix D: Model Scenarios

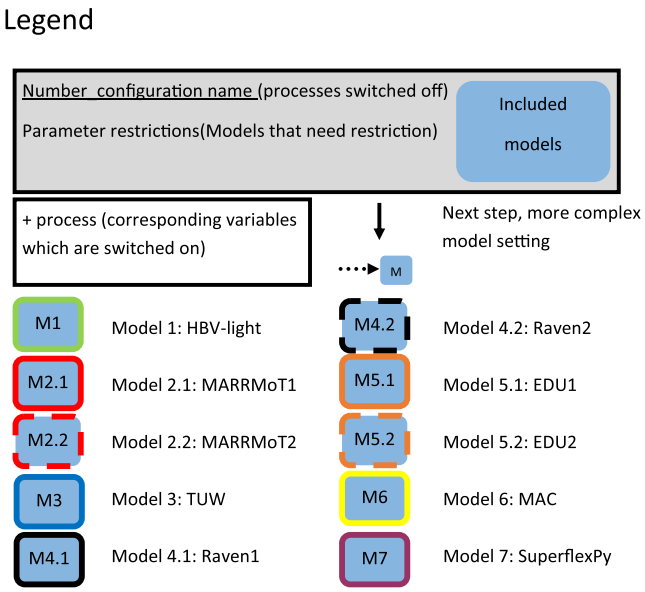


Figure D1. Legend model scenarios.

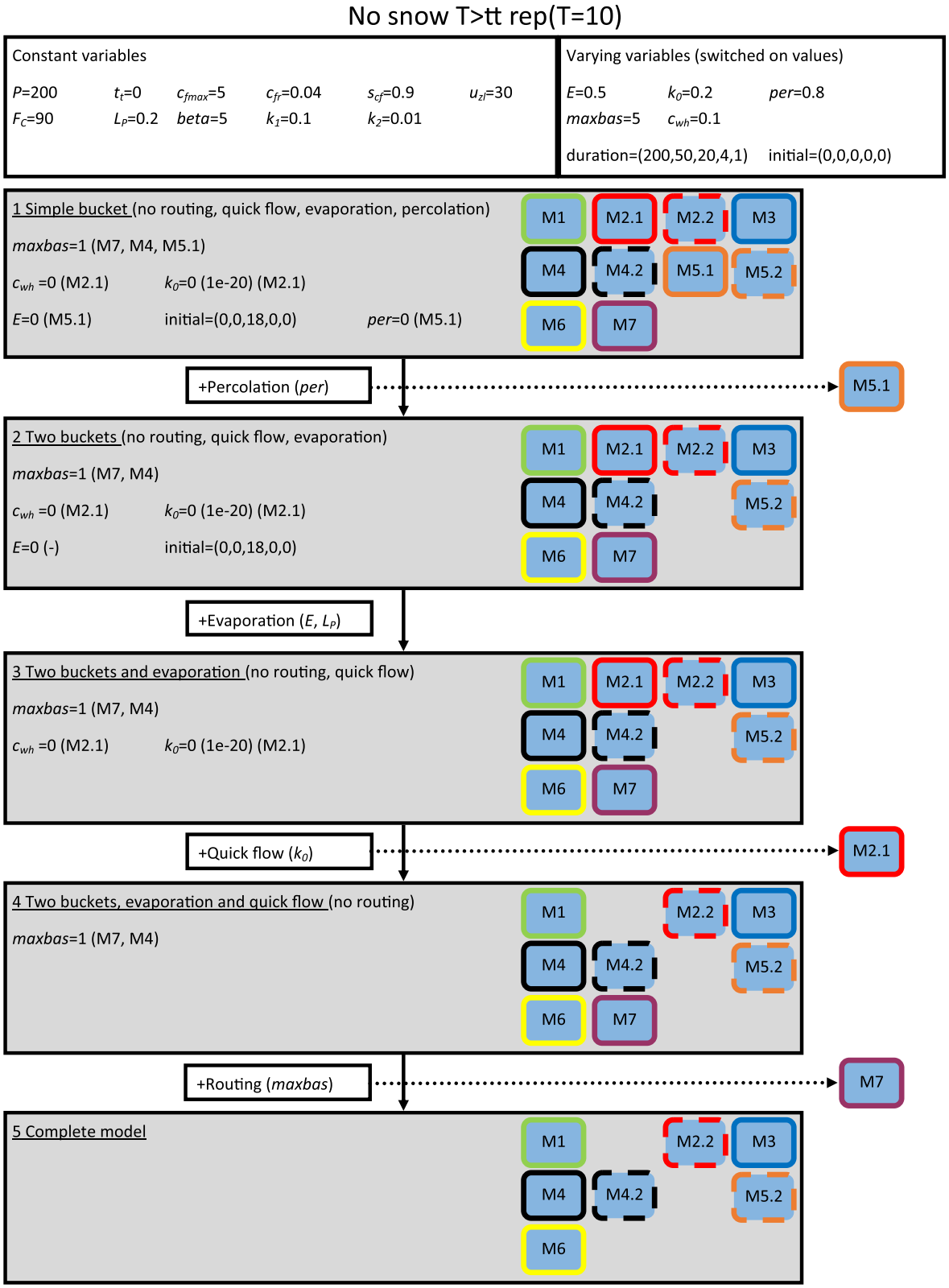


Figure D2. Graphical overview of parameter settings for the first five configurations (without snow) of the numerical comparison.

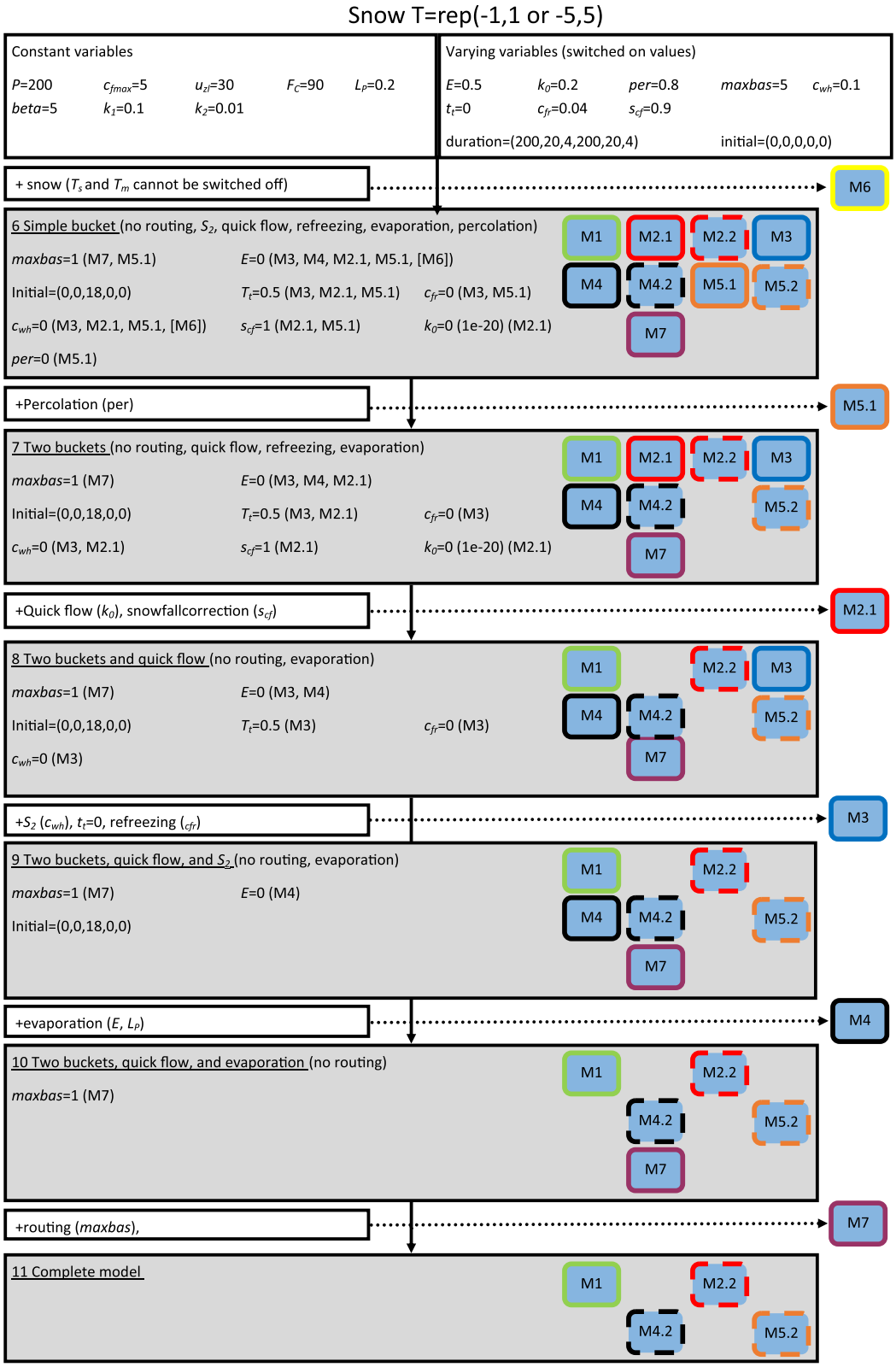


Figure D3. Graphical overview of parameter settings for the last six configurations (with snow) of the numerical comparison.

Appendix E: Forcing Data

Below, a description of the synthetic forcing data for the numerical comparison, is given. The synthetic forcing data of configuration 11 is also used for the mathematical comparison (2).

For the numerical comparison, synthetic forcing data is slightly different for the 11 configurations because it is used to switch snow processes and evaporation on/off. The synthetic data set is created such that it is as simple as possible while expressing all model processes. The precipitation events have an absolute value of 200 mm for all events. Those events are alternated by a recession period of 400 days. For each event, the intensity and duration change so that models' response to different intensities and duration is included. There are five precipitation events for the scenarios without snow processes and two times (with differing temperatures) the same three even ts for the scenarios that include snow processes. The event duration without snow processes are 200, 50, 20, 4, and 1 day with an intensity of 1, 4, 10, 50, and 200 mm/day. Only the (2X) 200, 20, and 4 days are used when snow processes are included.

The temperature is kept at 10°C when snow processes are switched off. When snow process are on, it alternates every $\frac{1}{4}$ precipitation duration between −1 and 1 for the first three precipitation events and between −5 and 5°C for the last three events to include different melting/refreezing rates. The alternation continues in the same frequency during the recession period where it still affects evaporation and melt rates. The potential evaporation is either kept at 0 or 0.5 mm/day depending on if evaporation is switched on.

The length of recession and spin-up time is a trade-off between starting all precipitation events with the same initial conditions and model run time. HBV-light was run for all different parameter and forcings combinations with a recession period of 300, 350, 400, and 500 days. Four-hundred days had little convergence compared to a recession period of 500 days with a maximum difference in discharge of 0.345 mm/day (spin-up data is shown in Table E1). The same 400 days were also set as spin-up period.

Table E1
Maximum Difference in Outflow for Different Recession Times for HBV-Light (mm/day) for the 11 Configurations in Part 2 (Numerical Implementation Differences)

Recession time	S_3 (mm)	1	2	3	4	5	6	7	8	9	10	11
300	2.2	0.001	0.014	0.334	1.314	0.849	0.001	0.028	0.016	0.01	0.981	0.644
350	0.8	0	0.007	0.0179	0.707	0.457	0	0.013	0.007	0.005	0.526	0.345
400	0.3	0	0.003	0.087	0.345	0.223	0	0.005	0.003	0.002	0.257	0.169

Note. The benchmark is HBV-light with a recession period of 500 days. Only the precipitation event and the shortest recession period are compared. The second column show the remainder of water in the soil moisture storage at the end of the first recession period. A recession time of 400 days is chosen as suitable.

HBV-light does not have initial conditions as model settings. Therefore, all models start with the same initial conditions as HBV-light. All storages start empty except for S_3 (soil moisture) which is set at $F_C \times L_P$ (wilting point). MAC cannot set the initial conditions, but the spin-up period is considered long enough to converge for MAC. The scenarios without evaporation start with a precipitation event of 20 mm/day for the first 20 days of the spin-up period to saturate S_3 and switch off the evaporation component for all precipitation events. The remaining 380 days is long enough to empty most storages. Only a small amount of <1 mm is left in S_5 (lower zone).

Appendix F: Calibration Convergence Acceleration

The calibration convergence was tested for 10 samples of HBV-light. Those results show to what extent the calibration is able to find the optimal solution. The convergence was found sufficient (results shown in result section) when applying an initial suggestion, parameter prioritization, and narrowing down the parameter range to accelerate calibration convergence. Those three measures are explained below.

The first measure to accelerate calibration convergence is an initial suggestion of parameter settings. This is given to one individual of the first generation for all model variants, other initial parameter values are chosen randomly within the parameter boundaries (Appendix G). This parameter setting is the same as used to generate the "observed data." This way, the GA starts from the optimal parameter settings for HBV-light and only compensates differences. The intricate parameters are switched off in this part.

Second, parameter prioritization is used to quicken calibration convergence. This is based on a local sensitivity analysis for the fast running models Raven1/2, TUW, MAC, and SuperflexPy. k_1 (normal flow linear coefficient), u_{zl} (quick flow threshold), *per* (percolation rate), F_C (infiltration splitter and evaporation parameter), s_{cf} (snow correction factor), and a_{thorn} (evaporation coefficient) were the most sensitive parameters. This is in agreement with previous sensitivity analysis (Ouyang, 2014; Seibert, 1997), which found k_1, *per*, F_C, and t_t the most sensitive. t_t is less sensitive because the variants are compared to a similar model structure instead of observed discharge. u_{zl} is more sensitive because of the sequential order in which HBV-light calculates S_4 (upper soil zone) fluxes, which is different than most other models. The sensitivity of k_1 and F_C could be linked to numerics in a similar way. Therefore, k_1, u_{zl}, *per*, and F_C are selected as calibration parameters. EDU1 uses per_{EDU} instead and MARRMoT1 replaces u_{zl} with *alpha* (quick flow is missing). The other parameters are fixed to the same values as in the previous parts and can be found in Tables A2 and A3.

Third, the parameter range was narrowed down after the local sensitivity analysis and after a calibration of 10 × 10 calibration run with all models. The final parameter ranges for the 50 × 10 calibration run are shown in Table G1. The calibrated parameters were checked to be not too close to the parameter range border.

Appendix G: Calibration Parameter Values

Table G1
The Calibrated Parameters are Shown for all Model Variants After 10 Iterations With a Population Size of 50

	KGE	*per*	u_{zl}	k_1	F_C	*alpha*	per_{EDU}
HBV-light	1	1.5	35	0.1	75	–	–
Lower boundary	–	1	20	0.05	50	−0.5	0.01
Upper boundary	–	2	50	0.2	100	1	0.05
MARRMoT1	0.345	1.42	-	0.169	84.0	−0.314	–
MARRMoT2	0.992	1.50	29	0.137	78.4	–	–
TUW	0.975	1.43	34.59	0.123	57.9	–	–
Raven1	0.957	1.73	33.88	0.134	59.5	–	–
Raven2	0.995	1.60	35.62	0.103	63.5	–	–
EDU1	0.897	-	43.56	0.067	96.1	–	0.0363
EDU2	0.979	1.70	36.03	0.103	79.1	–	–
MAC	0.931	1.42	30.37	0.145	87.8	–	–
SuperflexPy	0.992	1.43	29.29	0.127	78.0	–	–

Note. The first three rows show the parameter settings used to generate the observed data with HBV-light and the calibration boundaries. The fixed parameters that are not calibrated can be found in Tables A2 and A3.

Appendix H: Model Structure Differences

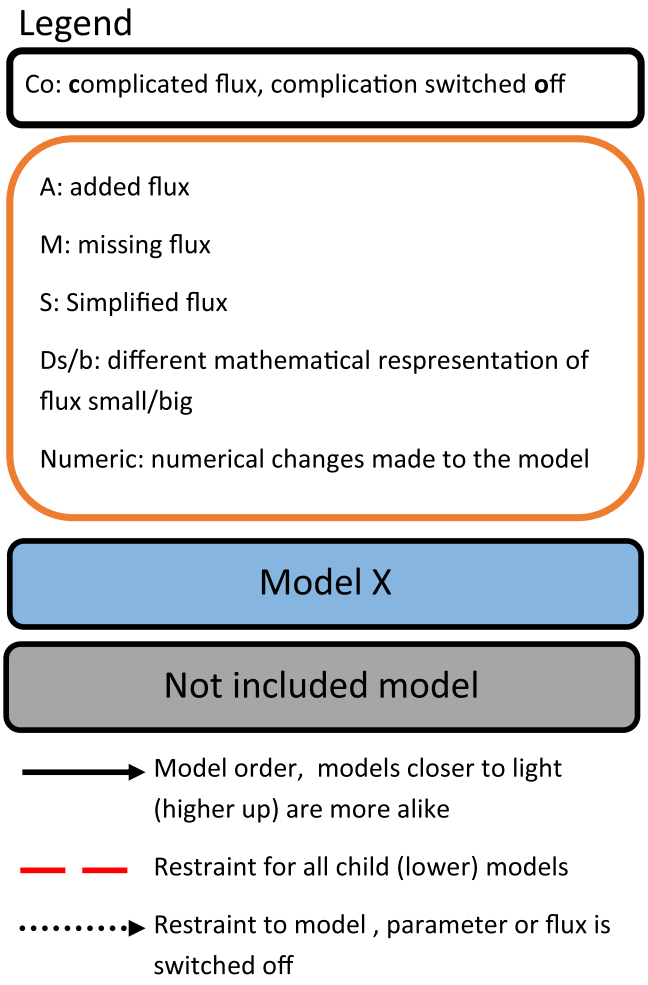


Figure H1. Legend model structure differences.

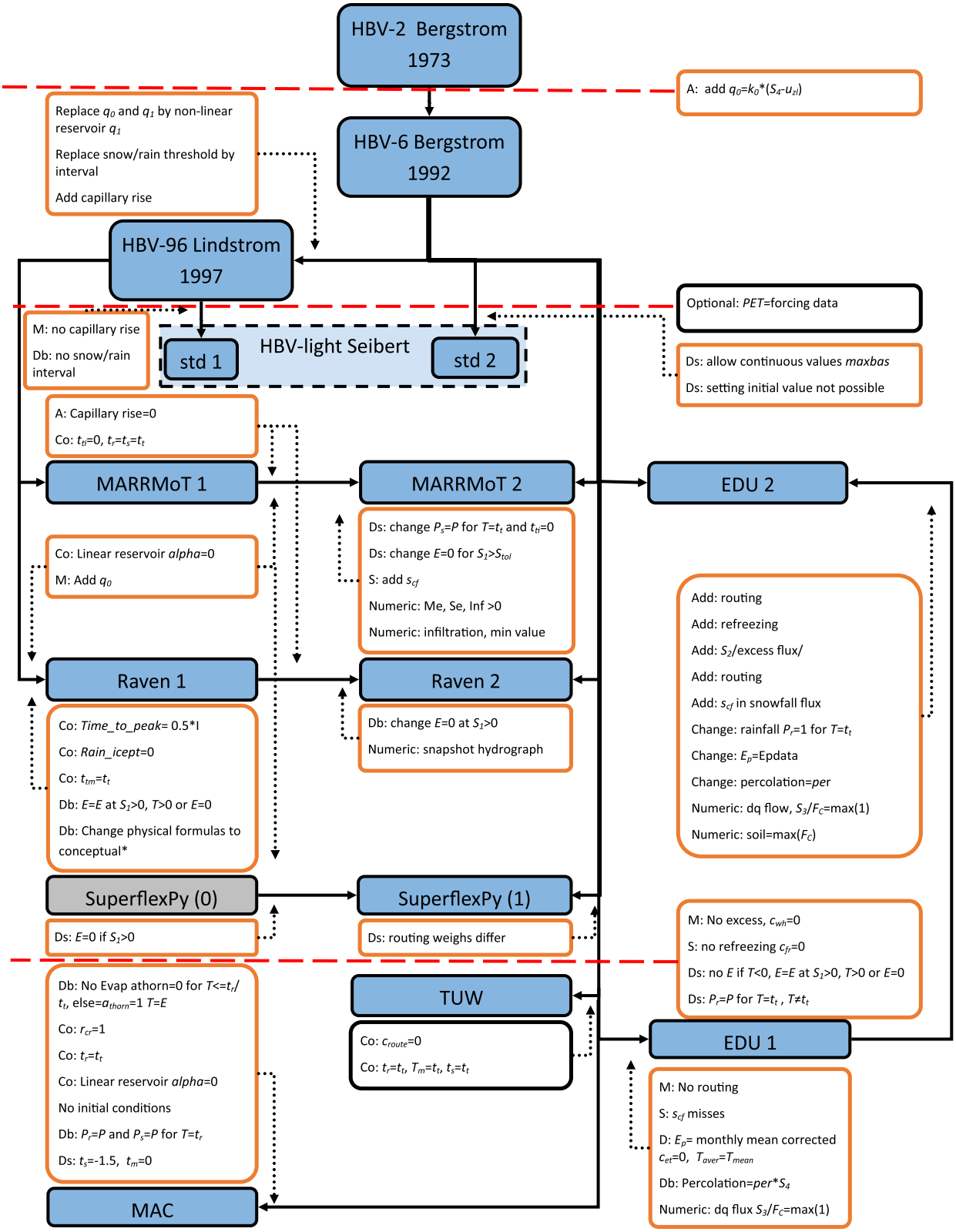


Figure H2. Overview of model differences and changes. Different HBV-versions are shown at the top of the figure to explain some of the differences in the other HBV-variants. The two standard version within HBV-light are indicated with std1 and std2. Std2 is used as the benchmark in this study. Parameter description is mostly consistent with the describing paper Table 1.

Appendix I: KGE Numerical Implementation

Table *I1. Kling-Gupta Efficiency for the Different Model Configurations of Increasing Model Complexity (Numerical Comparison), the Mathematical model is the Same for all Models Within One Configuration*

configuration	1	2	3	4	5	6	7	8	9	10	11	Legend
MARRMoT1	0.9599	0.9556	0.9711	0.4711	0.5993	0.9553	0.9466	0.6595	0.6544	0.7023	0.7120	>0.99
MARRMoT2	0.9599	0.9556	0.9711	0.8642	0.9033	0.9664	0.96215	0.8874	0.8901	0.9249	0.9434	>0.97
TUW	0.9580	0.9531	0.9523	0.8752	0.9135	0.9649	0.9600	0.9147	0.9297	0.8840	0.8791	>0.95
Raven1	0.9832	0.9797	0.9785	0.8528	0.9282	0.9908	0.9879	0.8930	0.8964	0.7682	0.7897	>0.9
Raven2	0.99999	0.99998	0.9949	0.9447	0.9572	0.99998	0.999996	0.9590	0.9602	0.9755	0.9825	>0.8
EDU1	0.99999	0.9818	0.5737	0.4638	0.4883	0.99998	0.9661	0.7496	0.7473	0.3398	0.2580	≤ 0.8
EDU2	0.99999	0.9948	0.9836	0.8482	0.9507	0.99998	0.9925	0.8875	0.8912	0.9351	0.9521	Excluded
MAC	0.99995	0.99998	-1.340	-0.2787	0.4703	0.8995	0.8787	0.8656	0.8814	-0.8913	-0.8611	
SUPERFLEX	0.9600	0.9557	0.9712	0.8644	0.9158	0.9665	0.9623	0.8877	0.8901	0.9251	0.9473	

[1]*Note.* The included models (which have the same mathematical model) are colored based on their mimicry performance and the excluded models are gray. The complexity increases with a higher number on top. MAC is unstable in configuration 3-4-5.

Appendix J: KGE Components Sampling

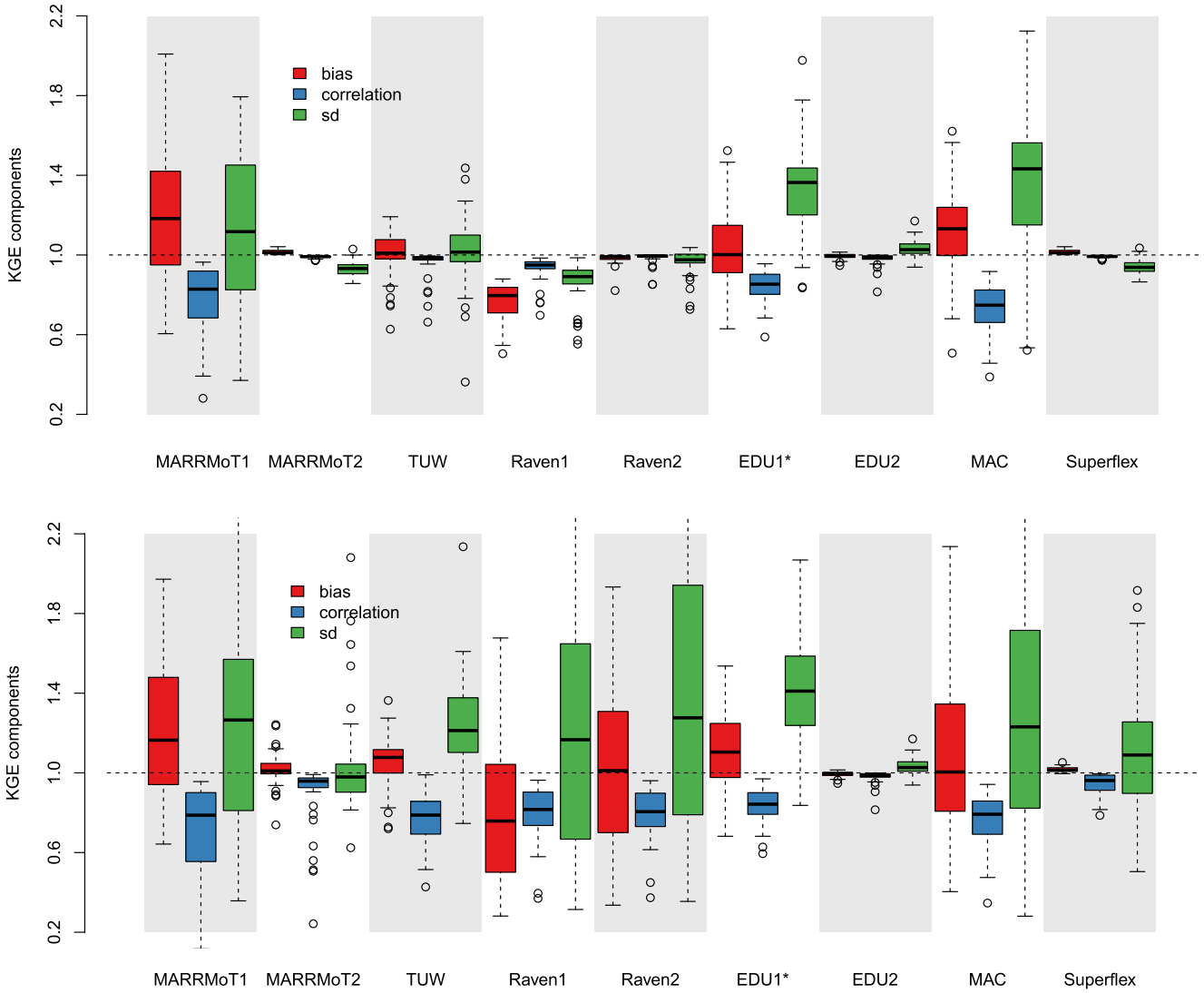


Figure J1. Kling-Gupta efficiency split up in the bias, correlation, and relative standard deviation for the mathematical comparison, conscious model use (top) and "off-the-shelf" model use sampling (bottom). $N = 50$ except for EDU1 which had 15 and 14 samples removed due to complex numbers in the results.

Figure J1. Kling-Gupta efficiency split up in the bias, correlation, and relative standard deviation for the mathematical comparison, conscious model use (top) and "off-the-shelf" model use sampling (bottom). N = 50 except for EDU1 which had 15 and 14 samples removed due to complex numbers in the results.

Data Availability Statement

All data was created with open source models using the parameter sets and forcing data as described in the original paper. The scripts that link the models in an R-environment and generate synthetic forcing data are available at http://www.hydroshare.org/resource/0908dd3550c947e695b4423ac72c7d41.

Acknowledgments
The authors would like to thank Peter la Follette for his help in writing scripts to build connections between models and Jan Seibert for his explanation about the HBV-light model. Furthermore, the authors would like to thank Anusha Mehta for her numerous textual feedback as a native English speaker and fruitful discussions about the paper its structure. Model source is indicated in Table 3.

References

Addor, N., & Melsen, L. A. (2019). Legacy, rather than adequacy, drives the selection of hydrological models. *Water Resources Research*, *55*(1), 378–390. https://doi.org/10.1029/2018WR022958

AghaKouchak, A., & Habib, E. (2010). Application of a conceptual hydrologic model in teaching hydrologic processes. *International Journal of Engineering Education*, *26*(4), 963–973.

Bancheri, M., Serafin, F., & Rigon, R. (2019). The representation of hydrological dynamical systems using Extended Petri Nets (EPN). *Water Resources Research*, *55*(11), 8895–8921. https://doi.org/10.1029/2019WR025099

Bauer, D. F. (1972). Constructing confidence sets using rank statistics. *Journal of the American Statistical Association*, *67*(339), 687–690. https://doi.org/10.1080/01621459.1972.10481279

Bergström, S. (1992). THE HBV MODEL - its structure and applications. *SMHI Reports Hydrology*.

Bergström, S. (2006). Experience from applications of the HBV hydrological model from the perspective of prediction in ungauged basins. *IAHS Publication*, *307*(97).

Bergström, S., & Forsman, A. (1973). Development of a conceptual deterministic rainfall-runoff model. *Hydrology Research*, *4*(3), 147–170. https://doi.org/10.2166/nh.1973.0012

Beven, K. J. (2011). *Rainfall-runoff modelling: The primer*. John Wiley & Sons.

Breuer, L., Huisman, J. A., Willems, P., Bormann, H., Bronstert, A., Croke, B. F. W., et al. (2009). Assessing the impact of land use change on hydrology by ensemble modeling (LUCHEM). I: Model intercomparison with current land use. *Advances in Water Resources*, *32*(2), 129–146. https://doi.org/10.1016/j.advwatres.2008.10.003

Carnell, R. (2020). *lhs: Latin Hypercube Samples*. Retrieved from https://CRAN.R-project.org/package=lhs

Clark, M. P., & Kavetski, D. (2010). Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes. *Water Resources Research*, *46*(10). https://doi.org/10.1029/2009WR008894

Clark, M. P., Kavetski, D., & Fenicia, F. (2011). Pursuing the method of multiple working hypotheses for hydrological modeling. *Water Resources Research*, *47*(9). https://doi.org/10.1029/2010WR009827

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., et al. (2015a). *The structure for unifying multiple modeling alternatives (SUMMA), version 1.0: Technical description. NCAR Tech. Note NCAR/TN-5141STR*.

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., et al. (2015b). A unified approach for process-based hydrologic modeling: 1. Modeling concept. *Water Resources Research*, *51*(4), 2498–2514. https://doi.org/10.1002/2015WR017198

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., et al. (2008). Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models. *Water Resources Research*, *44*(12). https://doi.org/10.1029/2007WR006735

Coxon, G., Freer, J., Lane, R., Dunne, T., Knoben, W. J., Howden, N. J., & Woods, R. (2019). DECIPHeR v1: Dynamic fluxEs and ConnectIvity for Predictions of HydRology. *Geoscientific Model Development*, *12*(6), 2285–2306. https://doi.org/10.5194/gmd-12-2285-2019

Craig, J. R., Brown, G., Chlumsky, R., Jenkinson, R. W., Jost, G., Lee, K., et al. (2020). Flexible watershed simulation with the Raven hydrological modelling framework. *Environmental Modelling & Software*, *129*, 104728. https://doi.org/10.1016/j.envsoft.2020.104728

Dal Molin, M., Kavetski, D., & Fenicia, F. (2020). SuperflexPy 1.2. 0: An open source python framework for building, testing and improving conceptual hydrological models. *Geoscientific Model Development Discussions*, 1–39.

Dowell, M., & Jarratt, P. (1972). The "Pegasus" method for computing the root of an equation. *BIT Numerical Mathematics*, *12*(4), 503–508. https://doi.org/10.1007/bf01932959

Fenicia, F., Kavetski, D., & Savenije, H. H. G. (2011). Elements of a flexible approach for conceptual hydrological modeling: 1. Motivation and theoretical development. *Water Resources Research*, *47*(11). https://doi.org/10.1029/2010WR010174

Girons Lopez, M., Vis, M. J. P., Jenicek, M., Griessinger, N., & Seibert, J. (2020). Complexity and performance of temperature-based snow routines for runoff modelling in mountainous areas in Central Europe. *Hydrology and Earth System Sciences Discussions*, 1–31. https://doi.org/10.5194/hess-2020-57

Gladish, D. W., Pagendam, D. E., Peeters, L. J. M., Kuhnert, P. M., & Vaze, J. (2018). Emulation engines: Choice and quantification of uncertainty for complex hydrological models. *Journal of Agricultural, Biological and Environmental Statistics*, *23*(1), 39–62. https://doi.org/10.1007/s13253-017-0308-3

Gupta, H. V., Clark, M. P., Vrugt, J. A., Abramowitz, G., & Ye, M. (2012). Towards a comprehensive assessment of model structural adequacy. *Water Resources Research*, *48*(8). https://doi.org/10.1029/2011WR011044

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, *377*(1–2), 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003

Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., & Arheimer, B. (2016). Most computational hydrology is not reproducible, so is it really science? *Water Resources Research*, *52*(10), 7548–7555. https://doi.org/10.1002/2016WR019285

Kampf, S. K., & Burges, S. J. (2007). A framework for classifying and comparing distributed hillslope and catchment hydrologic models. *Water Resources Research*, *43*(5). https://doi.org/10.1029/2006WR005370

Kirchner, J. W. (2006). Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology. *Water Resources Research*, *42*(3). https://doi.org/10.1029/2005WR004362

Knoben, W. J. M., Freer, J. E., Fowler, K. J. A., Peel, M. C., & Woods, R. A. (2019). Modular Assessment of Rainfall-Runoff Models Toolbox (MARRMoT) v1.2: An open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations. *Geoscientific Model Development*, *12*(6), 2463–2480. https://doi.org/10.5194/gmd-12-2463-2019

Knoben, W. J. M., Freer, J. E., & Woods, R. A. (2019). Technical note: Inherent benchmark or not? Comparing Nash-Sutcliffe and Kling-Gupta efficiency scores. *Hydrology and Earth System Sciences*, *23*(10), 4323–4331. https://doi.org/10.5194/hess-23-4323-2019

Knutti, R., Masson, D., & Gettelman, A. (2013). Climate model genealogy: Generation CMIP5 and how we got there. *Geophysical Research Letters*, *40*(6), 1194–1199. https://doi.org/10.1002/grl.50256

La Follette, P. T., Teuling, A. J., Addor, N., Clark, M., Jansen, K., & Melsen, L. A. (2021). Numerical daemons of hydrological models are summoned by extreme precipitation. *Hydrology and Earth System Sciences Discussions*. https://doi.org/10.5194/hess-2021-28

Leavesley, G. H., Markstrom, S. L., Restrepo, P. J., & Viger, R. J. (2002). A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modelling. *Hydrological Processes*, *16*(2), 173–187. https://doi.org/10.1002/hyp.344

Lindström, G., Johansson, B., Persson, M., Gardelin, M., & Bergström, S. (1997). Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology*, *201*(1–4), 272–288. https://doi.org/10.1016/S0022-1694(97)00041-3

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239–245. https://doi.org/10.1080/00401706.1979.10489755

Melsen, L. A., Addor, N., Mizukami, N., Newman, A. J., Torfs, P. J. J. F., Clark, M. P., et al. (2018). Mapping (dis)agreement in hydrologic projections. *Hydrology and Earth System Sciences*, *22*(3), 1775–1791. https://doi.org/10.5194/hess-22-1775-2018

Melsen, L. A., Torfs, P. J. J. F., Uijlenhoet, R., & Teuling, A. J. (2017). Comment on "Most computational hydrology is not reproducible, so is it really science?" by Christopher Hutton et al. *Water Resources Research*, *53*(3), 2568–2569. https://doi.org/10.1002/2016WR020208

Nijssen, B., Bennett, A., Clark, M., & Nearing, G. (2018). Using SUMMA for model mimicry: How do we define similarity between hydrologic models? *EGU general assembly conference abstracts* (Vol. 20, p. 10936).

Ouyang, S., Puhlmann, H., Wang, S., von Wilpert, K., & Sun, O. J. (2014). Parameter uncertainty and identifiability of a conceptual semi-distributed model to simulate hydrological processes in a small headwater catchment in Northwest China. *Ecological Processes*, *3*(1), 14. https://doi.org/10.1186/s13717-014-0014-9

Parajka, J., Merz, R., & Blöschl, G. (2007). Uncertainty and multiple objective calibration in regional water balance modelling: Case study in 320 Austrian catchments. *Hydrological Processes*, *21*(4), 435–446. https://doi.org/10.1002/hyp.6253

Peters, N. E., Freer, J., & Beven, K. (2003). Modelling hydrologic responses in a small forested catchment (Panola Mountain, Georgia, USA): A comparison of the original and a new dynamic TOPMODEL. *Hydrological Processes*, *17*(2), 345–362. https://doi.org/10.1002/hyp.1128

Remmers, J. O. E., Teuling, A. J., & Melsen, L. A. (2020). Can model structure families be inferred from model output? *Environmental Modelling & Software*, *133*, 104817. https://doi.org/10.1016/j.envsoft.2020.104817

Samuel, J., Coulibaly, P., & Metcalfe, R. A. (2011). Estimation of continuous streamflow in Ontario ungauged basins: Comparison of regionalization methods. *Journal of Hydrologic Engineering*, *16*(5), 447–459. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000338

Scibek, J., & Allen, D. M. (2006). Modeled impacts of predicted climate change on recharge and groundwater levels. *Water Resources Research*, *42*(11). https://doi.org/10.1029/2005WR004742

Scrucca, L. (2013). GA: A package for Genetic Algorithms in R. *Journal of Statistical Software*, *53*(4), 1–37. https://doi.org/10.18637/jss.v053.i04

Seibert, J. (1997). Estimation of parameter uncertainty in the HBV model. *Nordic Hydrology*, *28*(4), 247–262. https://doi.org/10.2166/nh.1998.15

Seibert, J. (2005). *HBV light version 2 user's manual*. Department of Earth Sciences, Uppsala University.

Seibert, J., & Vis, M. J. P. (2012). Teaching hydrological modeling with a user-friendly catchment-runoff-model software package. *Hydrology and Earth System Sciences*, *16*(9), 3315–3325. https://doi.org/10.5167/uzh-67295

Siegel, S., & Castellan, N. J. (1981). *Nonparametric statistics for the behavioral sciences*. McGraw-HiU Book Company.

Skidmore, A. (2002). Taxonomy of environmental models in the spatial sciences. *Environmental Modelling with GIS and Remote Sensing*, 8–25. https://doi.org/10.1201/9780203302217.ch2

Teuling, A. J., de Badts, E. A. G., Jansen, F. A., Fuchs, R., Buitink, J., Hoek van Dijke, A. J., & Sterling, S. M. (2019). Climate change, reforestation/afforestation, and urbanization impacts on evapotranspiration and streamflow in Europe. *Hydrology and Earth System Sciences*, *23*(9), 3631–3652. https://doi.org/10.5194/hess-23-3631-2019

Uhlenbrook, S., Seibert, J., Leibundgut, C., & Rodhe, A. (1999). Prediction uncertainty of conceptual rainfall-runoff models caused by problems in identifying model parameters and structure. *Hydrological Sciences Journal*, *44*(5), 779–797. https://doi.org/10.1080/02626669909492273

Wagener, T., Sivapalan, M., Troch, P. A., McGlynn, B. L., Harman, C. J., Gupta, H. V., et al. (2010). The future of hydrology: An evolving science for a changing world. *Water Resources Research*, *46*(5). https://doi.org/10.1029/2009WR008906

Weiler, M., & Beven, K. (2015). Do we need a community hydrological model? *Water Resources Research*, *51*(9), 7777–7784. https://doi.org/10.1002/2014WR016731

Xu, C.-Y., Seibert, J., & Halldin, S. (1996). Regional water balance modelling in the NOPEX area: Development and application of monthly water balance models. *Journal of Hydrology*, *180*(1–4), 211–236. https://doi.org/10.1016/0022-1694(95)02888-9

Zektser, I., & Loaiciga, H. A. (1993). Groundwater fluxes in the global hydrologic cycle: Past, present and future. *Journal of Hydrology*, *144*(1–4), 405–427. https://doi.org/10.1016/0022-1694(93)90182-9